

# 无人机编队飞行中的纯方位无源定位问题的研究

## 摘要

无人机作为一种新兴科技逐步被投入到各种领域,实际运用中也会产生各种问题。无人机群在空中遂行编队飞行时,为了避免受到外界信号的干扰,应最大限度保持电磁静默,即尽量减少向外界发射电磁波信号。为保持队形,机群拟使用纯方位无源定位的方法调整无人机的位置,由编队中的几架无人机发射信号、其余无人机被动接受信号并从中提取出方向信息对自身进行定位。本文通过建立数学模型来给出无人机的具体定位方法和位置调整方案。

**针对问题一的第一问**,首先通过几何作图,判断位于圆心的无人机(FY00)和编队中另 2 架(位置无偏差且编号已知)无人机发射信号能否实现其余位置略有偏差的无人机定位,即能确定位置有偏差的无人机的唯一位置。然后分别通过平面建系法和解三角形法建立模型并得出位置有偏差的无人机的位置参数。

**针对问题一的第二问**,在第一问研究的基础上容易发现,如果只添加一架编号未知的无人机发射信号,则方程的个数少于未知数个数,无法确定被动接受信号无人机的唯一位置。于是考虑添加两架编号位置的无人机发射信号,分别建立多变量混合整数非线性规划模型和相交圆方程模型,最终实现了无人机的唯一定位。因此,除 FY00 和 FY01 外,还需要 2 架无人机发射信号,才能实现无人机的有效定位。

**针对第一题的第三问**,观察表 1 中的数据,我们发现编号为 FY01 的无人机的半径正好是 100,且角度为 0。为了简化问题,我们不妨每次选择固定编号为 FY00 的无人机和 FY01 的无人机,再选择圆周上另一架无人机作为信号源向其余无人机发射信号进行调整。我们采用基于贪婪算法的位置调整模型。结果使得 9 架无人机均匀分布在半径为 100 的圆周上。

**针对第二题**,我们每次定位好一批距离较近的无人机,令这一批无人机作为新的信号源向剩余的无人机发射信号,以此类推。首先选中编号为 FY01, FY02, FY03 作为最初的信号源,让这三架无人机定位至均匀分布在一个圆上,再定位剩余的无人机。对于给定某列,我们设计出了,可以计算出这一列上,每一架无人机应该收到来自于哪几架无人机的信号的算法程序。

**关键词:** 无人机 无源定位 混合整数规划模型 贪婪算法 递推算法

## 一、问题重述

### 1.1 问题背景

无人机作为一种新兴科技逐步被投入到各种领域，实际运用中也会产生各种问题。无人机群在空中遂行编队飞行时，为了避免受到外界信号的干扰，应最大限度保持电磁静默，即尽量减少向外界发射电磁波信号。在不发射电磁波信号的情况下，为保持队形，机群拟使用纯方位无源定位的方法调整无人机的位置，由编队中的几架无人机发射信号、其余无人机被动接受信号并从中提取出方向信息对自身进行定位。

### 1.2 问题重述

题目约定了接收信号的无人机所接收到的方向信息为：该无人机与任意两架发射信号无人机连线之间的夹角，即若无人机接受了若干其他无人机的信号，即可得到它与任意两架发送信号无人机连线的夹角。

问题一给出了一个由十架无人机组成的圆形编队，其中一架（编号 FY00）位于圆心，其中 9 架无人机（编号 FY01~FY09）均匀分布在某一圆周上。无人机可以自身感知高度，从而保持整个编队位于同一高度。要求根据这一情景回答下列问题：

(1) 位于圆心的无人机（FY00）和编队中另 2 架无人机发射信号，其余位置略有偏差的无人机被动接收信号。发射信号的无人机位置无偏差且编号已知，要求根据上述条件挖掘无人机之间的几何关系，建立被动接收信号无人机的定位模型并求解。

(2) 某位置略有偏差的无人机接收到编号为 FY00 和 FY01 的无人机发射的信号，另接收到编队中若干编号未知的无人机发射的信号，发射信号的无人机位置无偏差。该问对第一问的问题进行的条件变形，要求求出除 FY00 和 FY01 外能实现无人机有效定位的最少无人机数量。

(3) 该问是前两问的进一步深化，要求在建立了定位模型的基础上，给出位置有偏差无人机的位置调整方案：每次选择编号为 FY00 的无人机和圆周上最多 3 架无人机遂行发射信号，其余无人机根据接收到的方向信息，调整到理想位置，通过多次调整，使得 9 架无人机最终均匀分布在某个圆周上。利用表 1 给出的数据，仅根据接收到的方向信息来调整无人机的位置，请给出具体的调整方案。

表 1 无人机的初始位置

无人机编号	极坐标(m, °)
0	(0, 0)
1	(100, 0)
2	(98, 40.10)
3	(112, 80.21)
4	(105, 119.75)

5	(98, 159.86)
6	(112, 199.96)
7	(105, 240.07)
8	(98, 280.17)
9	(112, 320.28)

问题二给出了一个由 15 架无人机组成的锥形编队，对问题一的情景进行了改变，要求在熟悉和理解第一问的基础上做到迁移，仍利用无源定位来给出具体的位置调整方案，使任意两架相邻无人机之间的距离相等。

## 二、问题分析

### 2.1 问题一（1）的分析

问题给出了一个由十架无人机组成的圆形编队，要求根据已知条件挖掘无人机之间的几何关系，建立被动接收信号无人机的定位模型并求解。根据题意，我们先判断位于圆心的无人机（FY00）和编队中另 2 架（位置无偏差且编号已知）无人机发射信号能否实现其余位置略有偏差的无人机定位。由圆的对称性，我们可设其中一架发射信号的无人机为 FY01，不失一般性。由题设可知，FY 的轨迹既在以 FY00-FY01 为弦的一段圆弧上（不包括端点），也在以 FY00-FYK1 为弦的一段圆弧上（不包括端点），由于两段圆弧有公共点且不相切，因此两段圆弧必将交与另一点，该点即为 FY 的唯一位置。我们可以建立平面直角坐标系，将已知的几何关系坐标化，求解未知参数，从而得到 FY 的位置。还可以结合图形，运用解三角形的相关知识列出有关未知参数的方程并求解，也能得到 FY 的位置。

### 2.2 问题一（2）的分析

问题一第二问要求求出除 FY00 和 FY01 外能实现无人机有效定位的最少无人机数量。在第一问研究的基础上容易发现，如果只添加一架编号未知的无人机发射信号，则方程的个数少于未知数个数，无法确定被动接受信号无人机的唯一位置。于是考虑添加两架编号未知的无人机发射信号。由于被动接受信号的无人机的位置与原本的位置略有偏差，我们可以假设无人机的偏离位置在以原位置为圆心，以  $0.02R$  为半径的圆内，建立多变量混合整数规划模型，以 FY 的位置和未知的两台发射信号的无人机编号为决策变量，以角度余弦值之差的平方和的最小值为目标进行规划，得出最优解，此为法一。在第一问中，我们用相交圆来确定 FY 的位置，在此问中可作进一步推广。由于存在四个未知数，我们可以构造 FY 的四个圆轨迹求交点，即等价于联立四条圆的方程求解，此为法二。

### 2.3 问题一（3）的分析

观察表 1 中的数据，我们发现编号为 FY01 的无人机的半径正好是 100，且角度为 0。为了简化问题，我们不妨每次选择固定编号为 FY00 的无人机和 FY01

的无人机,再选择圆周上另一架无人机作为信号源向其余无人机发射信号进行调整,可建立基于贪婪算法的位置调整模型。

2.4 问题二的分析

我们由前面的问题推断得出,对于每一台需要调整的无人机,只需要接收到另外三架无人机发出的信号即可完成定位。准确定位需要三台作为信号源的无人机位置也准确,但如果无人机只能进行信号接收和发送,其中的一个功能的话,初始的三架信号源位置将不会均匀分布在一个圆周上,所以所有的无人机都是可以进行信号的收发。此外,固定了三台无人机向剩余的无人机发射信号,如果某一无人机本身距离那三架信号源较远,那么移动改变相同的角度时,与距离较近的无人机相比,移动的距离会更大,有可能发生碰撞。故我们每一次定位好一批距离较近的无人机时,令这一批无人机作为新的信号源向剩余的无人机发射信号,以此类推。

三、模型假设

- (1) 假设被动接受信号的无人机接受到的方位信息是准确无误的。
- (2) 假设无人机调整位置的时间可以忽略不计。
- (3) 假设整个飞行过程中所有无人机均位于同一高度,即在同一个平面上。
- (4) 假设无人机在飞行过程中没有受到外界信号的干扰,无特殊情况发生。
- (5) 假设圆形编队的半径是已知量。

四、符号说明

符号	符号说明
$R$	圆形编队的半径大小
$\overline{FY00 - FY01}$	FY00 和 FY01 连接的线段
$\overline{FY - FY01}$	FY 和 FY01 连接的线段
$\overline{FY - FY00}$	FY 和 FY00 连接的线段
$\overline{FY - FY0K}$	FY 和 FYK1 连接的线段
$\overline{FY00 - FYK1}$	FY00 和 FYK1 连接的线段
$\alpha_1$	$\overline{FY - FY00}$ 与 $\overline{FY - FY0K}$ 的夹角
$\alpha_2$	$\overline{FY - FY00}$ 与 $\overline{FY - FY01}$ 的夹角

$\alpha_3$	$\overline{FY - FY00}$ 与 $\overline{FY - FY1K}$ 的夹角
$\alpha_4$	$\overline{FY - FY0K}$ 与 $\overline{FY - FY01}$ 的夹角
$k+1$	发射信号无人机的编号
$i+1$	被动接受信号无人机的编号
$(\rho, \theta)$	被动接受信号无人机的极坐标
$(x, y)$	被动接受信号无人机的直角坐标

## 五、模型的建立与求解

### 5.1 问题一（1）模型的建立和求解

#### 5.1.1 定位解的存在性和唯一性检验

由圆的对称性可知圆上的无人机地位等价，因此为了简化模型，不妨设其中一架发射信号的无人机为 FY01，不失一般性。由题设可知，位置有偏差的无人机（设为 FY）与 FY00 和 FY01 连线的夹角已知，且 FY00 和 FY01 距离一定，为圆的半径 R，所以 FY 的轨迹为以  $\overline{FY00 - FY01}$  为弦的一段圆弧上（不包括端点），同理 FY 与 FY00 和 FY0K 连线的夹角已知，所以 FY 的轨迹在以  $\overline{FY00 - FY0K}$  为弦的一段圆弧上（不包括端点）。由于两段圆弧有公共点且不相切，因此两段圆弧必将交与另一点，该点即为 FY 的唯一位置。设圆形编队的半径大小为 R， $\overline{FY - FY00}$ 与 $\overline{FY - FY01}$  的夹角为  $\alpha_1$ ，设 $\overline{FY - FY00}$ 与 $\overline{FY - FY0K}$  的夹角为  $\alpha_2$ ，发射信号无人机的编号为 k+1（k 为自然数），被动接受信号无人机的直角坐标为 (x, y)。由于原本九架无人机均匀分布在圆周上，因此相邻两架无人机之间的夹角相等，记为  $360^\circ / 9 = 40^\circ$ ，记为  $\alpha$ 。根据题目以及所设，可作出如下示意图：

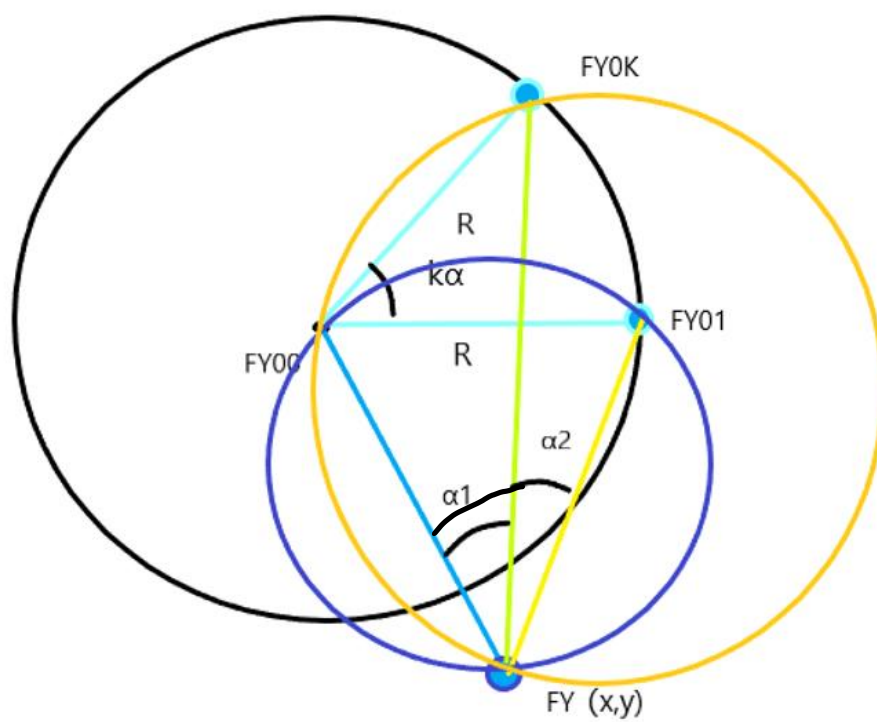
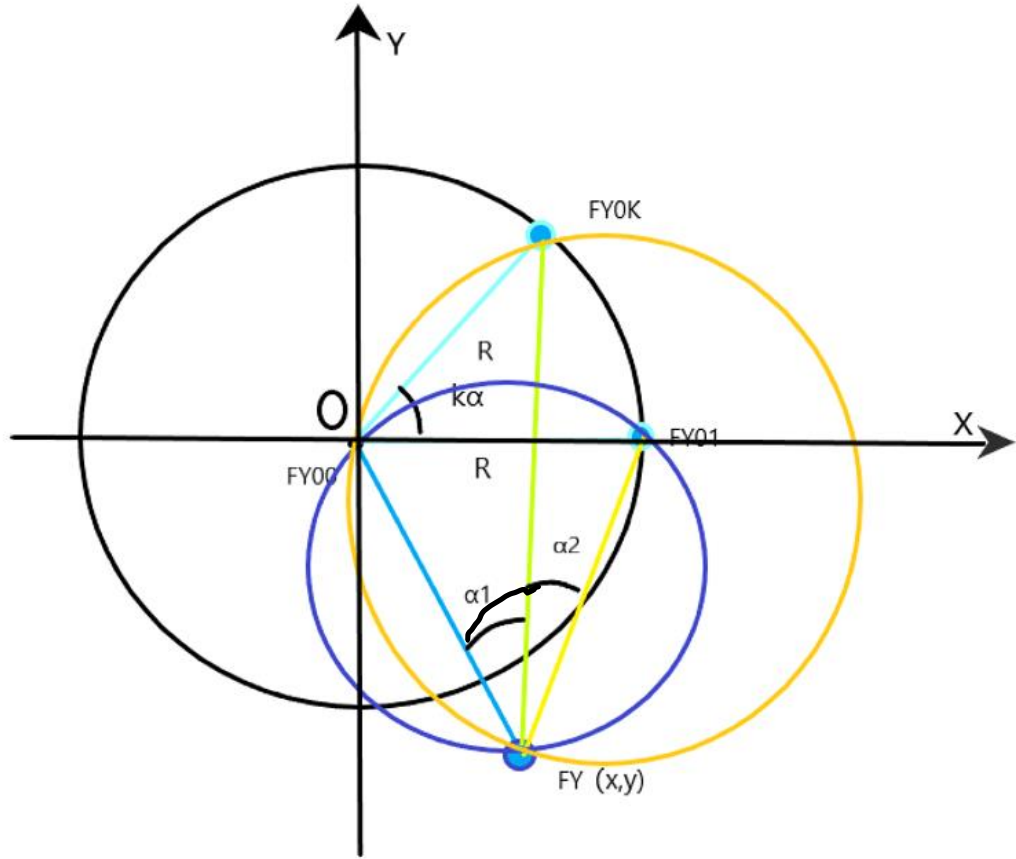


图 1：定位解的存在性和唯一性检验示意图

### 5.1.2 平面建系法定位无人机

如图所示，以  $FY00$  为原点， $FY00$  指向  $FY01$  方向为  $x$  轴正方向建立平面直角坐标系得， $FY00(0,0)$ 、 $FY01(R,0)$ 、 $FY(x,y)$ 、 $FY0K(R\cos k\alpha, R\sin k\alpha)$



在三角形FY - FY00 - FY0K中对  $\alpha_1$  运用余弦公式，在三角形FY - FY01 - FY00中对  $\alpha_2$  运用余弦公式，得

$$\begin{aligned}\cos \alpha_2 &= \frac{(FY00 - FY)^2 + (FY - FY0K)^2 - (FY00 - FY0K)^2}{2(FY00 - FY)(FY - FY0K)} \\ &= \frac{x^2 + y^2 + (x - R)^2 + y^2 - R^2}{2\sqrt{x^2 + y^2}\sqrt{(x - R)^2 + y^2}}\end{aligned}\quad (1)$$

$$\begin{aligned}\cos \alpha_1 &= \frac{(FY00 - FY)^2 + (FY - FY01)^2 - (FY00 - FY01)^2}{2(FY00 - FY)(FY01 - FY)} \\ &= \frac{x^2 + y^2 + (x - R \cos k\alpha)^2 + (y - R \sin k\alpha)^2 - R^2}{2\sqrt{x^2 + y^2}\sqrt{(x - R \cos k\alpha)^2 + (y - R \sin k\alpha)^2}}\end{aligned}\quad (2)$$

$$\text{由于 FY 的位置略有偏差，不妨设 } 0.98R \leq \sqrt{x^2 + y^2} \leq 1.02R \quad (3)$$

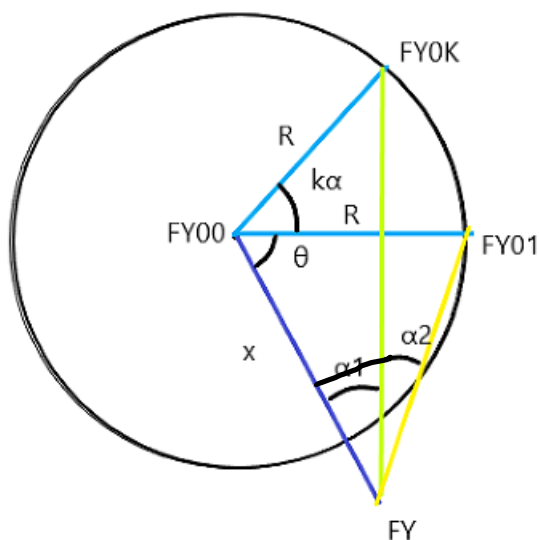
根据题意  $\alpha_1$ ,  $\alpha_2$ ,  $\alpha$ ,  $k$  已知，只有  $x$ ,  $y$  两个未知数，联立 (1) (2) (3)，该方程组是一个非线性方程组，可利用 MATLAB 中自带的 fsolve 函数通过最小二乘法求解得到数值解，从而实现定位。

### 5.1.3 解三角形法定位无人机

由于发射信号无人机和接受信号无人机的相对位置不同，在不同情况下构建的三角形以及相应的角度不同，下面根据  $k$  的大小进行分类讨论。

(1) 当  $k=2, 3$  时，如下图所示：

对三角形  $FY - FY00 - FY0K$  和三角形  $FY - FY00 - FY01$  分别运用正弦定理得：



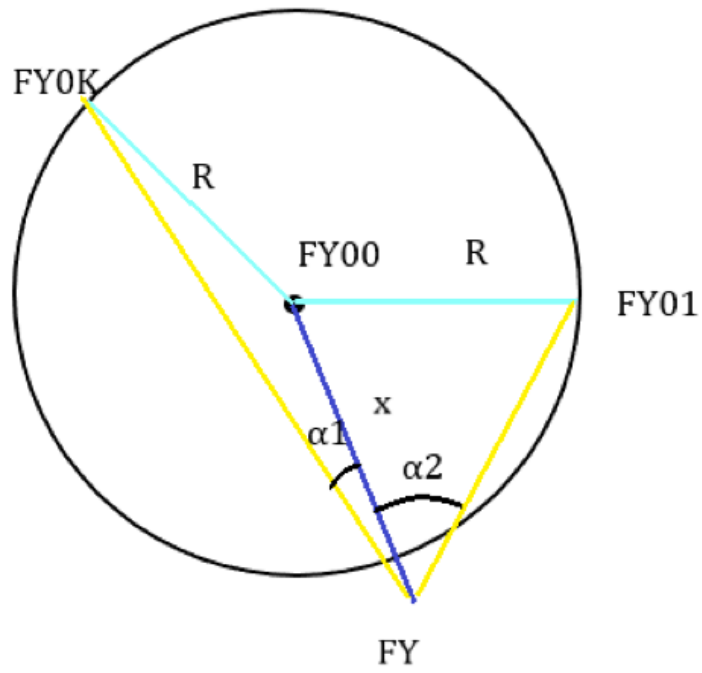
$$\begin{cases} \frac{\sin \alpha_2}{R} = \frac{\sin[\pi - (2\pi - \theta) - \alpha_2]}{x} \\ \frac{\sin \alpha_1}{R} = \frac{\sin[\pi - (2\pi - \theta + k\alpha + \alpha_1)]}{x} \end{cases}$$

联立解得：

$$\begin{cases} \theta = \arctan \left( \frac{\sin \alpha_1 \cdot \sin(\alpha_1 + k\alpha) - \sin \alpha_1 \sin \alpha_2}{\sin \alpha_2 \cos(\alpha_2 + k\alpha) - \sin \alpha_1 \cos \alpha_2} \right) \\ x = \frac{R \sin(\alpha_2 - \theta)}{\sin \alpha_2} \end{cases}$$

(2) 当  $k=4, 5$  且  $k\alpha > \theta$  时，如下图所示：





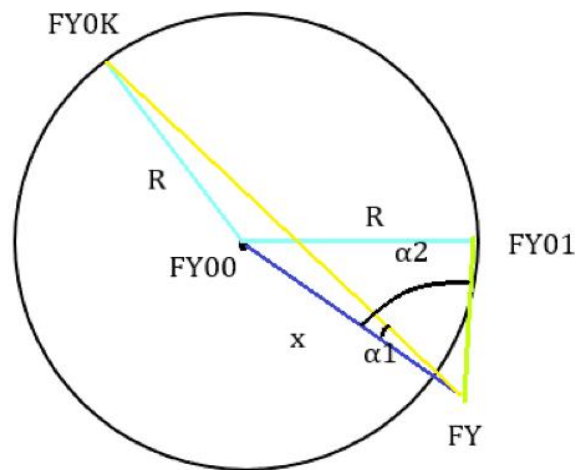
对三角形FY – FY00 – FY0K和三角形FY – FY00 – FY0K分别运用正弦定理得：

$$\begin{cases} \frac{\sin \alpha_1}{R} = \frac{\sin[\pi - \alpha_1 - (2\pi - \theta)]}{x} \\ \frac{\sin \alpha_2}{R} = \frac{\sin[\pi - \alpha_2 - (\theta - k\alpha)]}{x} \end{cases}$$

联立解得：

$$\begin{cases} \theta = \arctan \left[ \frac{\sin \alpha_2 \sin(\alpha_1 - k\alpha) - \sin \alpha_1 \sin \alpha_2}{\sin \alpha_2 \cos(\alpha_1 - k\alpha) - \sin \alpha_1 \cos \alpha_2} \right] \\ x = \frac{R \cdot \sin(\alpha_2 - \theta)}{\sin \alpha_2} \end{cases}$$

(3) 当  $k=4, 5$  且  $k\alpha < \theta$  时，如下图所示：



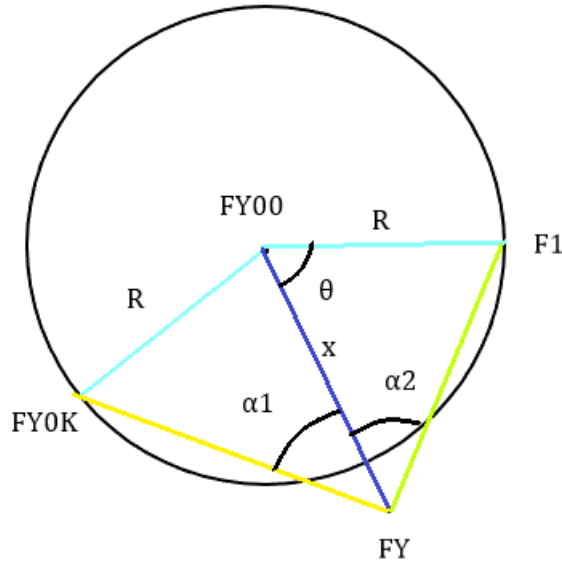
对三角形FY – FY00 – FY0K和三角形FY – FY00 – FY0K分别运用正弦定理得：

$$\begin{cases} \frac{\sin \alpha_2}{R} = \frac{\sin(\pi - \theta - \alpha_2)}{x} \\ \frac{\sin \alpha_1}{R} = \frac{\sin[\pi - (\theta - k\alpha) - \alpha_2 - \alpha_1]}{x} \end{cases}$$

联立解得：

$$\begin{cases} \theta = \arctan \left( \frac{\sin \alpha_2 \sin (\alpha_1 + k\alpha) - \sin \alpha_1 \sin \alpha_2}{\sin \alpha_1 \cos (\alpha_1 + k\alpha) + \sin \alpha_2 \cos \alpha_1} \right) \\ x = \frac{R \cdot \sin (\theta + \alpha_2)}{\sin \alpha_2} \end{cases}$$

(4) 当k=6, 7, 8, 如下图所示：



对三角形FY – FY00 – FY0K和三角形FY – FY00 – FY0K分别运用正弦定理得：

$$\begin{cases} \frac{\sin \alpha_2}{R} = \frac{\sin(\pi - \theta - \alpha_2)}{x} \\ \frac{\sin \alpha_1}{R} = \frac{\sin[\pi - \alpha_1 - (k\alpha - \theta)]}{x} \end{cases}$$

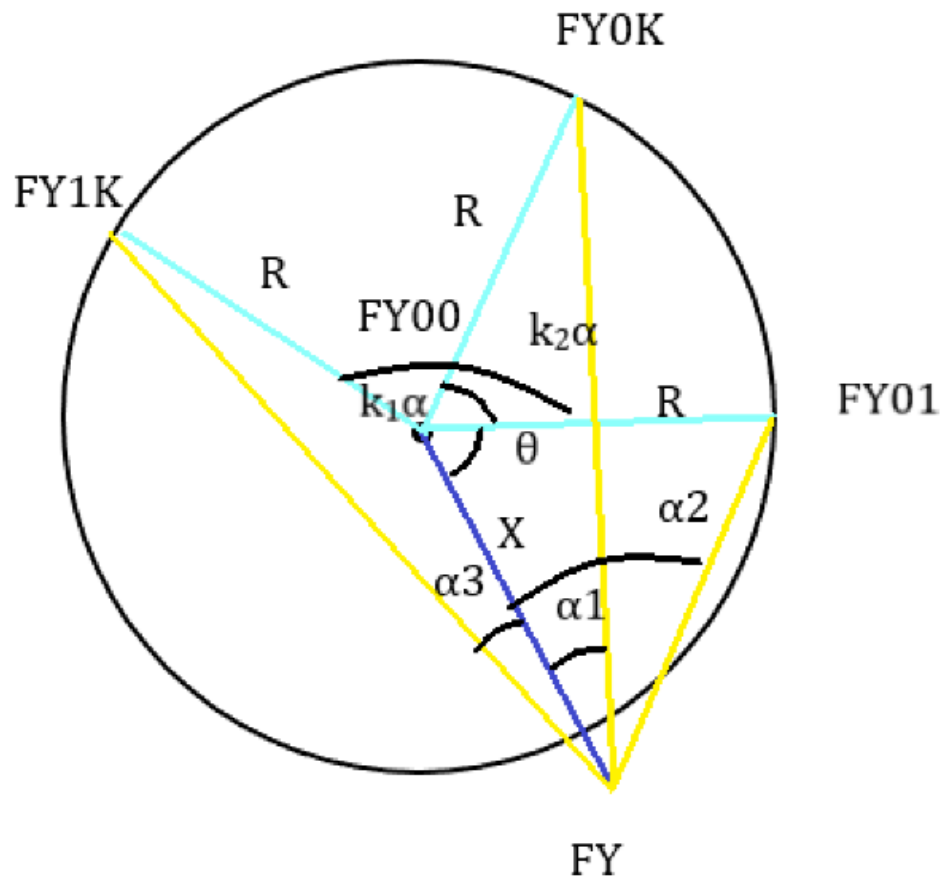
联立解得：

$$\begin{cases} \theta = \arctan \left[ \frac{\sin \alpha_2 \sin (\alpha_1 + k\alpha) - \sin \alpha_1 \sin \alpha_2}{\sin \alpha_2 \cos (\alpha_1 + k\alpha) - \sin \alpha_1 \cos \alpha_2} \right] \\ x = \frac{R \cdot \sin (\alpha_2 - \theta)}{\sin \alpha_2} \end{cases}$$

## 5.2 问题一（2）模型的建立和求解

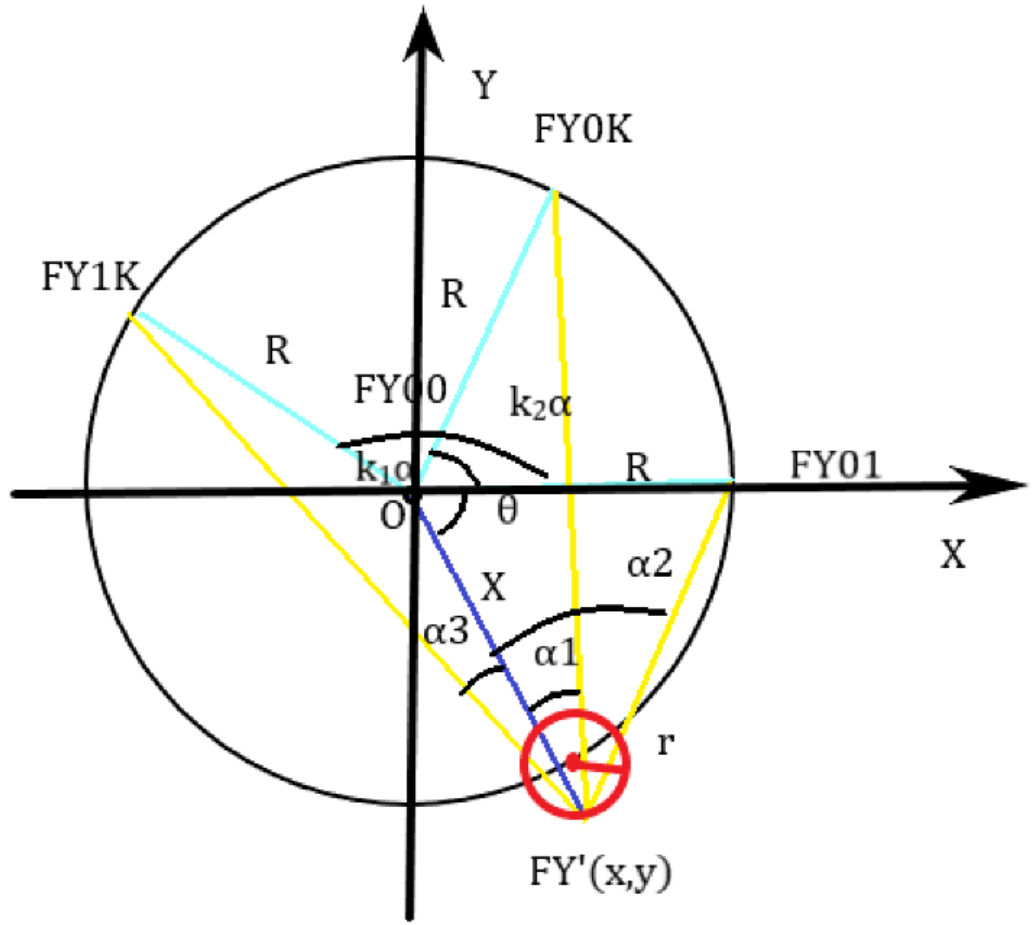
在第一问研究的基础上容易发现，如果只添加一架编号未知的无人机发射信号，则方程中的k为未知数，导致方程的个数少于未知数个数，无法确定被动接

受信号无人机的唯一位置。于是考虑添加两架编号未知的无人机发射信号，记为 FY0K 和 FY1K，设这两架无人机编号为  $k_1+1$  和  $k_2+1$ ，不妨设  $k_1 < k_2$  则可画出如下示意图：



### 5. 2. 1 混合整数非线性规划模型的构建与求解

由于被动接受信号的无人机位置偏差不大，我们假设其与原位置距离相差不超过  $r=0.02R$ ，在此圆形区域内对无人机进行定位。设区域内的点为 FY'，与 FY1K、FY0K、FY01、FY00 连线，记  $\overline{FY' - FY00}$  与  $\overline{FY' - FY0K}$  的夹角为  $\beta_1$ ，记  $\overline{FY' - FY00}$  与  $\overline{FY' - FY01}$  的夹角为  $\beta_2$ ，记  $\overline{FY' - FY00}$  与  $\overline{FY' - FY1K}$  的夹角为  $\beta_3$ 。若 FY' 即为 FY，则相应角度的差值应该为 0。设被动接受信号的无人机编号为 i。建立如下图所示的平面直角坐标系，得 FY00 (0,0)、FY01(R,0)、FY'(x, y)、FY0K ( $R\cos k_1\alpha, R\sin k_1\alpha$ )、FY1K ( $R\cos k_2\alpha, R\sin k_2\alpha$ )。



将几何约束用坐标运算表示出来，可建立如下的混合整数非线性规划模型：

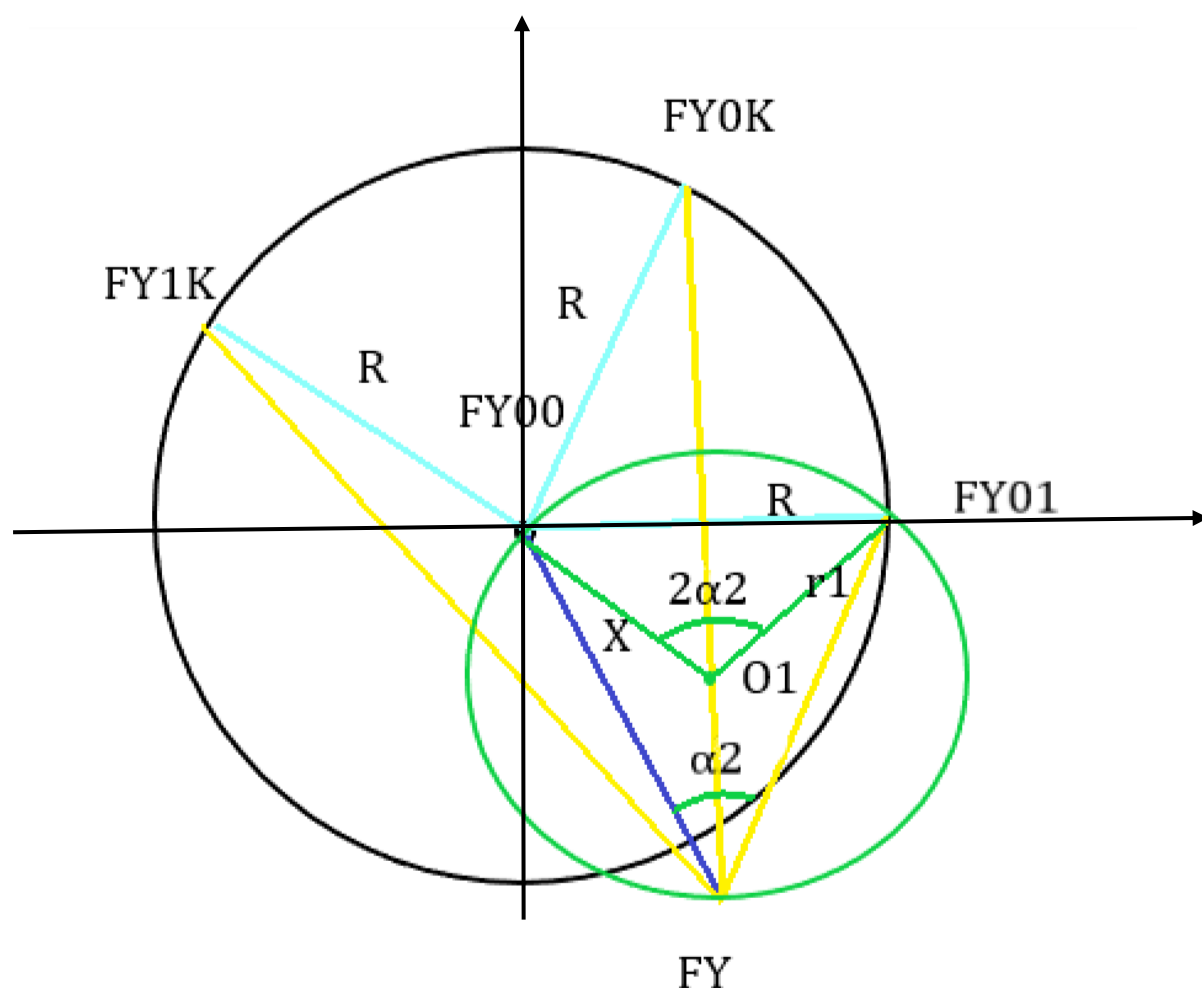
$$\begin{aligned}
 & \text{Min } (\cos \beta_1 - \cos \alpha_1)^2 + (\cos \beta_2 - \cos \alpha_2)^2 + (\cos \beta_3 - \cos \alpha_3)^2 \\
 & \text{s. t } \left\{ \begin{array}{l}
 i, k_1, k_2 \in N^* \\
 1 \leq k_1 < k_2 \leq 8 \\
 r = 0.02R \\
 (x - R \cos i\alpha)^2 + (y - R \sin i\alpha)^2 \leq r^2 \\
 i \neq k_1, i \neq k_2 \\
 \cos \beta_1 = \frac{x^2 + y^2 + (x - R \cos k_1 \alpha)^2 + (y - R \sin k_1 \alpha)^2 - R^2}{2\sqrt{x^2 + y^2} \sqrt{(x - R \cos k_1 \alpha)^2 + (y - R \sin k_1 \alpha)^2}} \\
 \cos \beta_2 = \frac{x^2 + y^2 + (x - R)^2 + y^2 - R^2}{2\sqrt{x^2 + y^2} \sqrt{(x - R)^2 + y^2}} \\
 \cos \beta_3 = \frac{x^2 + y^2 + (x - R \cos k_2 \alpha)^2 + (y - R \sin k_2 \alpha)^2 - R^2}{2\sqrt{x^2 + y^2} \sqrt{(x - R \cos k_2 \alpha)^2 + (y - R \sin k_2 \alpha)^2}} \\
 2 \leq i \leq 8
 \end{array} \right.
 \end{aligned}$$

决策变量为  $x, y, k_1, k_2$ ，其他量均为已知量。利用 MATLAB 自带的 OPTI toolbox 可进行求解。由于  $x, y$  具有唯一性，理论上能得到最优解。因此，除 FY00 和 FY01 外，还需要 2 架无人机发射信号，才能实现无人机的有效定位。

### 5.2.2 相交圆定位模型的构建与求解

在第一问中,我们用相交圆来确定了 FY 的位置,在此问中可作进一步推广。由于存在四个未知数,我们可以构造 FY 的四个圆轨迹求交点,即等价于联立四条含四个未知数的圆的方程求解。已知两个定点以及不同在一条直线上的一个动点与定点连线之间的夹角为定值可确定动点的轨迹为一个圆或两段圆弧(两个定点为圆弧交点),两个定点为圆的直径或圆弧上的弦。因此想要构造圆轨迹,需要先知道两个定点的位置以及不同在一条直线上的一个动点与定点连线之间的夹角。

以 FY00 为圆心, FY00 指向 FY01 方向为 x 轴建立平面直角坐标系,得到各点的坐标: FY00 (0,0)、FY01(R,0)、FY(x, y)、FY0K ( $R\cos\alpha_1, R\sin\alpha_1$ )、FY1K ( $R\cos\alpha_2, R\sin\alpha_2$ )下面通过选取四组不同的定点和夹角构造四个不同的圆轨迹。  
(1)选取 FY00 和 FY01 为定点,  $\alpha_2$  为夹角构建圆  $O_1$ , 设圆心为  $O_1(a_1, b_1)$ , 半径为  $r_1$ 。



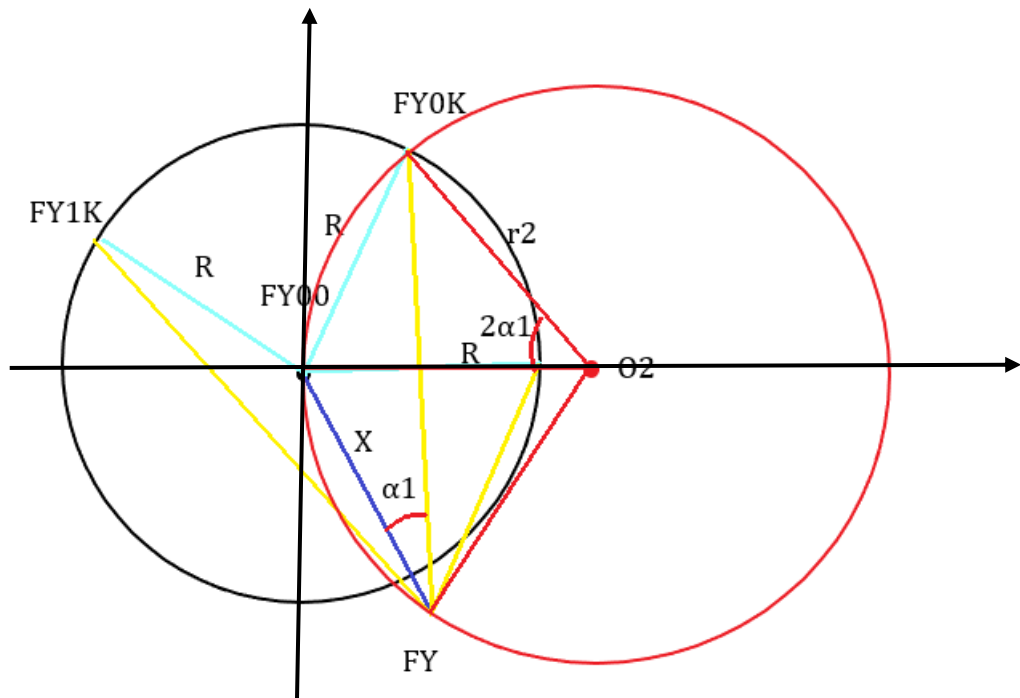
由于 FY00 和 FY01 在该圆上,结合圆周角定理和垂径定理,可列出如下方程组:

$$\begin{cases} a_1^2 + b_1^2 = r_1^2 \\ (a_1 - R)^2 + b_1^2 = r_1^2 \\ R = 2r_1 \cos 2\alpha_2 \end{cases}$$

解得：

$$\begin{cases} r_1 = \frac{R}{2\cos \alpha_2} \\ a_1 = \frac{R}{2} \\ b_1 = \frac{R}{2} \cdot \tan \alpha_2 \end{cases}$$

(2) 选取 FY00 和 FY0K 为定点， $\alpha_1$  为夹角构建圆  $O_2$ ，设圆心为  $O_2(a_2, b_2)$ ，半径为  $r_2$ 。



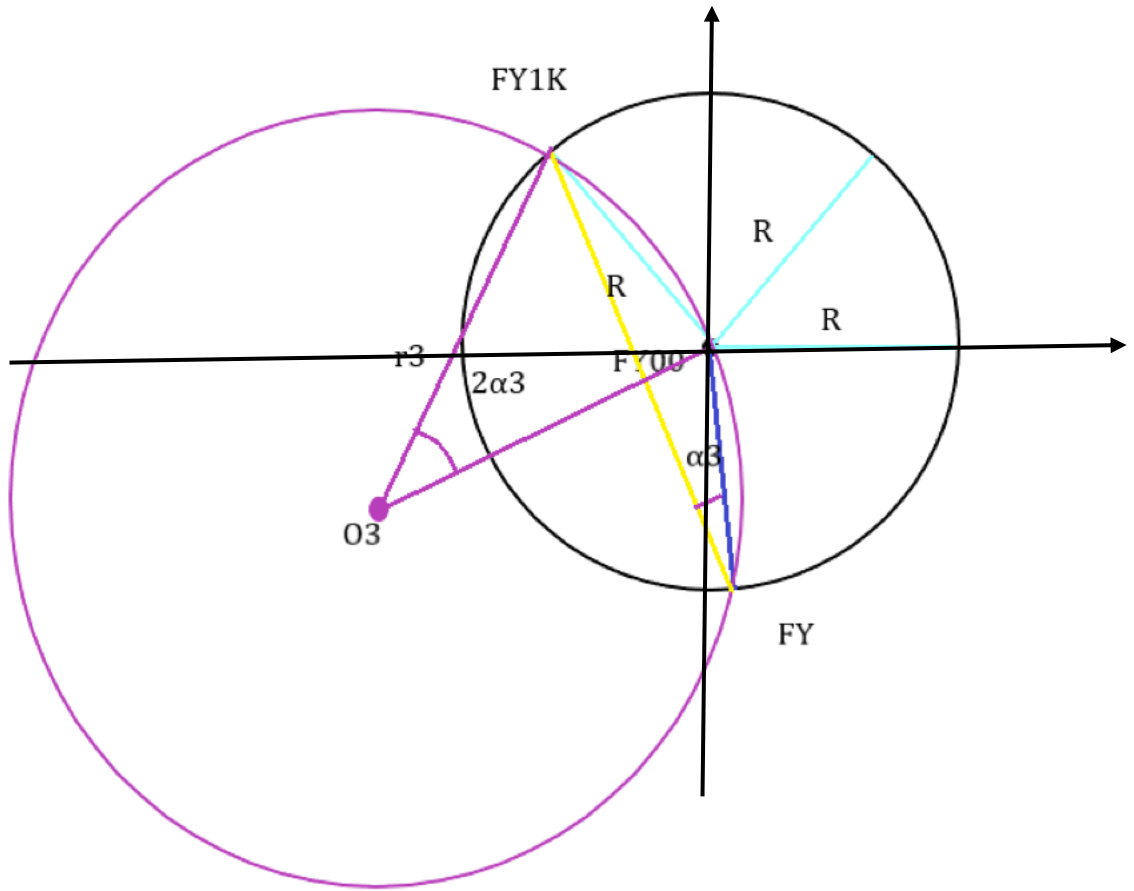
由于 FY00 和 FY0K 在该圆上，结合圆周角定理和垂径定理，可列出如下方程组：

$$\begin{cases} (a_2 - 0)^2 + (b_2 - 0)^2 = r_2^2 \\ (a_2 - R \cos k_1 \alpha)^2 + (b_2 - R \sin k_1 \alpha)^2 = r_2^2 \\ R = 2r_2 \cos \alpha_1 \end{cases}$$

解得：

$$\begin{cases} r_2 = \frac{R}{2\cos \alpha_1} \\ b_2 = \sqrt{r_2^2 + R^2 \left( \frac{1}{2} - \frac{\sin(k_1 \alpha)^2}{4} \right)} - \frac{R}{2} \sin k_1 \alpha \\ a_2 = \frac{R}{2\cos k_1 \alpha} + b_2 \tan k_1 \alpha, \end{cases}$$

(3)选取 FY00 和 FY1K 为定点， $\alpha_3$  为夹角构建圆  $O_3$ ，设圆心为  $O_3(a_3, b_3)$ ，半径为  $r_3$ 。



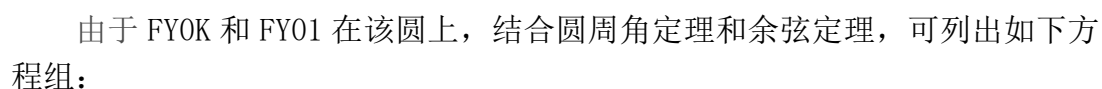
由于 FY00 和 FY1K 在该圆上，结合圆周角定理和垂径定理，可列出如下方程组：

$$\begin{cases} (a_3 - 0)^2 + (b_3 - 0)^2 = r_3^2 \\ (a_3 - R \cos k_2 \alpha)^2 + (b_3 - R \sin k_2 \alpha)^2 = r_3^2 \\ R = 2r_3 \cos \alpha_3 \end{cases}$$

解得：

$$\begin{cases} r_3 = \frac{R}{2 \cos \alpha_3} \\ b_3 = \sqrt{r_3^2 + R^2 \left( \frac{1}{2} - \frac{\sin(k_2 \alpha)^2}{4} \right)} - \frac{R}{2} \sin k_2 \alpha \\ a_3 = \frac{R}{2 \cos k_2 \alpha} + b_3 \tan k_2 \alpha, \end{cases}$$

(4)选取 FY0K 和 FY1K 为定点， $\alpha_4$  为夹角构建圆  $O_4$ ，设圆心为  $O_4(a_4, b_4)$ ，半径为  $r_4$



$$\begin{cases} (a_4 - R \cos k_1 \alpha)^2 + (b_4 - R \sin k_1 \alpha)^2 = r_4^2 \\ (x_4 - R)^2 + y_4^2 = r_4^2 \\ (R - R \cos k_1 \alpha)^2 + (1 - R \sin k_1 \alpha)^2 = 2r_4^2(1 - \cos 2\alpha_4) \end{cases}$$

$$\left\{ \begin{array}{l} r_4^2 = \frac{R^2(1 - \cos k_1\alpha)^2 + (1 - \sin(k_1\alpha)R)^2}{2(1 - \cos 2\alpha_4)} \\ b_4 = \frac{1 - \cos k_1\alpha}{\sin k_1\alpha} a_4, \\ a_4 = \frac{R}{(1 + \frac{(1 - \cos k_1\alpha)^2}{\sin(k_1\alpha)^2})} + \sqrt{\frac{R^2}{(1 + \frac{(1 - \cos k_1\alpha)^2}{\sin(k_1\alpha)^2})} - \frac{r_4^2}{(1 + \frac{(1 - \cos k_1\alpha)^2}{\sin(k_1\alpha)^2})}} \end{array} \right.$$
$$\begin{cases} (x - a_1)^2 + (y - b_1)^2 = r_1^2 \\ (x - a_2)^2 + (y - b_2)^2 = r_2^2 \\ (x - a_3)^2 + (y - b_3)^2 = r_3^2 \\ (x - a_4)^2 + (y - b_4)^2 = r_4^2 \end{cases}$$



该方程组中  $\alpha_1, \alpha_2, \alpha_3$  为测量得到的已知量,  $x, y, k_1, k_2$  为所要求的未知量, 四条方程相互独立。该方程组是一个非线性方程组, 可利用 MATLAB 中自带的 `fsolve` 函数通过最小二乘法求解得到数值解, 从而实现定位。因此, 除 FY00 和 FY01 外, 还需要 2 架无人机发射信号, 才能实现无人机的有效定位。

### 5.3 问题一 (3) 模型的建立及求解

#### 5.3.1 贪婪算法原理

贪婪算法 (Greedy algorithm) 是一种对某些求最优解问题的更简单、更迅速的设计技术。用贪婪法设计算法的特点是一步一步地进行, 常以当前情况为基础根据某个优化测度作最优选择, 而不考虑各种可能的整体情况, 它省去了为找最优解要穷尽所有可能而必须耗费的大量时间, 它采用自顶向下, 以迭代的方法做出相继的贪心选择, 每做一次贪心选择就将所求问题简化为一个规模更小的子问题, 通过每一步贪心选择, 可得到问题的一个最优解。

#### 5.3.2 无人机具体调整方案

观察表 1 中的数据, 我们发现编号为 FY01 的无人机的半径正好是 100, 且角度为 0。为了简化问题, 我们不妨每次选择固定编号为 FY00 的无人机和 FY01 的无人机, 再选择圆周上另一架无人机作为信号源向其余无人机发射信号进行调整。我们采用基于贪婪算法的位置调整模型。结果使得 9 架无人机均匀分布在半径为 100 的圆周上。

贪婪算法的原理是在通过每次寻找局部最优解来近似逼全局最优解, 所以在每一次调整前, 希望找到除 FY00, FY01 以外的另一架与目标位置偏差最小的无人机用作信号源。由表 1 我们可以得知除 FY00, FY01 以外, 每架无人机都有不同程度的半径和角度的偏差, 所以我们定义了一个偏差指数  $Q$  来衡量该编号的飞机与目标位置的偏差程度。该偏差指数  $Q$  的具体定义如下:

$$Q = \left| \frac{\rho_i - R}{R} \right| + \left| \frac{\theta_{i1} - \theta_{i2}}{\theta_{i2}} \right| (i = 1, 2, \dots, 9)$$

其中,  $\rho_i$  为经过  $i$  次调整后的飞机的极径,  $\theta_{i1}$  为经过  $i$  次调整后的飞机的相对于  $x$  轴方向的角度,  $\theta_{i2}$  为编队在圆周上均匀分布时无人机与  $x$  轴的夹角

每一次的调整分为一下几步:

##### 1. 寻找无人机:

我们希望找出权重指数最小的那架飞机。我们对于信号有信号选择指的定义: 每一次选择权重最小的无人机与 FY00, FY01 一起作为信号源。

2. 对比角度: 首先我们知道当 9 架无人机均匀分布于目标圆周上时, 不同编号的无人机与 FY00, FY01 之间感知到的夹角大小是不同的。用第一步时选择了 FY02 举例, FY03 到 FY09 每一架无人机都能从 FY00, FY01, FY02 收到三个与编号相对应的正确的角, 设为  $\beta_1, \beta_2$ ; 实际均匀分布在圆上的角设为  $\alpha_1, \alpha_2$ 。此时每一架无人机都用现在实际收到的三个夹角大小与正确的标准夹角大小进行对比, 从而得知自身有偏差, 并进行移动。

3. 调整无人机位置的决策: 首先由题干得知, 无人机基于收到的角度进行位置调整的移动, 那么相对于圆心移动的距离和角度, 就和实际接收到的三个角度和目标位置的理想的三个角度有关。那么我们提出两条方程:

$$\rho_{i+1} = \rho_i + m1(\max\{\beta1, \beta2\} - \min\{\beta1, \beta2\} - (\max\{\alpha1, \alpha2\} - \min\{\alpha1, \alpha2\}))$$

$$(i = 1, 2, \dots, n)$$

$$\theta_{i+1, 1} = \theta_{i, 1} + m2(\max\{\beta1, \beta2\} - \max\{\alpha1, \alpha2\} - (\min\{\beta1, \beta2\} - \min\{\alpha1, \alpha2\}))$$

$$(i = 1, 2, \dots, n)$$

方程中的 $m1, m2$ 是设置的调整系数，相当于衡量了无人机调整位置的能力（精度）。例如： $m$ 越小时，无人机每一次移动的距离就小，减少了过度调整的可能，但是增加了算法的迭代次数 $i$ 。

第二条方程是衡量相对于圆心移动的角度大小，与实际得到的两个夹角都有关。计算调整之后每一台飞机的位置。有了第三步的方程，我们可以得知其余无人机相对于表 1 所移动的距离。相当于得到了一个新的表 1，再进行下一次的贪婪计算。我们通过调整  $n1, n2$  和迭代的次数，最终无人机会近似均匀分布于一个基于 FY00, FY01 的圆周上。

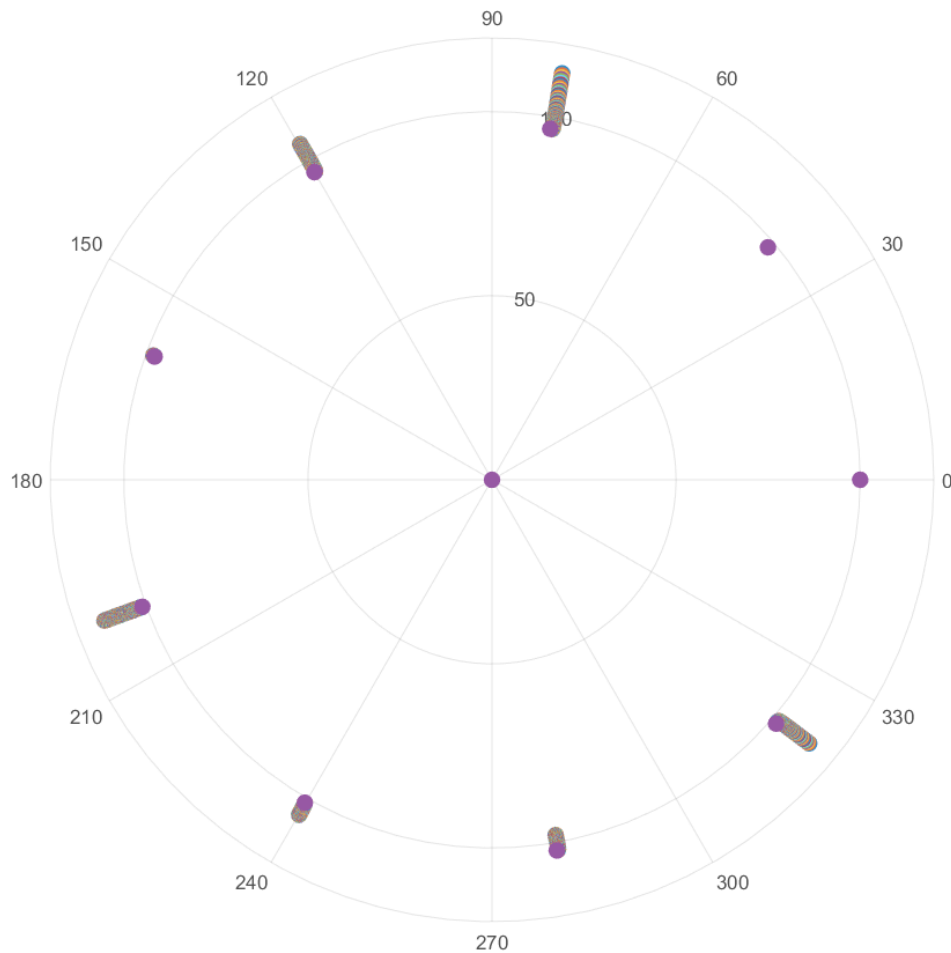


图 2：程序算法迭代了 500 次取得的结果

如图是程序算法迭代了 500 次取得的结果，其中  $k_1$  取 0.5， $k_2$  取 0.0005。可以看到，其中的点已经在每一次迭代中向期望的半径为 100 的圆周移动，当点经过半径为 100 的圆周，则可近似看成该算法对该点有解。

### 5.3 问题二模型的建立及求解

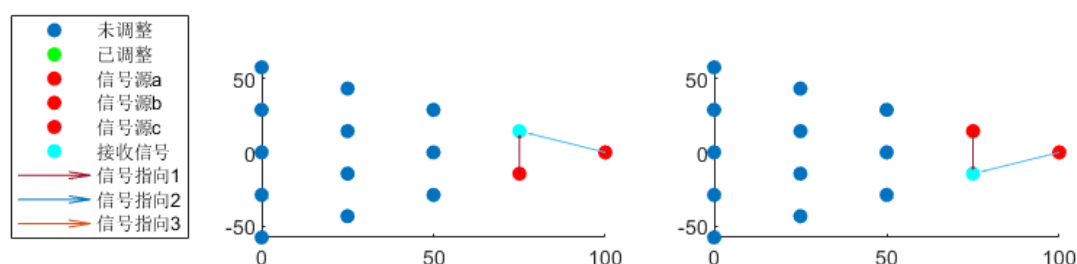
我们由前面的问题推断得出，对于每一台需要调整的无人机，只需要接收到另外三架无人机发出的信号即可完成定位。准确定位需要三台作为信号源的无人机位置也准确，初始的三架信号源位置将不会均匀分布在一个圆周上，所以所有的无人机都是可以进行信号的收发。此外，固定了三台无人机向剩余的无人机发射信号，如果某一无人机本身距离那三架信号源较远，那么移动改变相同的角度时，与距离较近的无人机相比，移动的距离会更大，产生较大的误差。故我们每一次定位好一批距离较近的无人机时，不会选择单一信号源的策略，而是令这新的调整好后的一批无人机作为信号源向剩余的无人机发射信号，以此类推。所以，在飞机接受临近飞机的信号时调整的误差会更小。

#### 5.3.1 选取信号源定位剩余的无人机

在飞机沿  $y$  轴的两个方向没有同时存在相邻的飞机时选取临近的两架飞机作为信号源。

基于优化步数的考虑，在飞机非（1）情况时选取三架飞机作为信号源。

在信号源选择策略开始前，至少需要有三架相邻的飞机作为最初的准确信号源，所以，我们选中编号为 FY01，FY02，FY03 作为最初的信号源，故需要首先让这三架无人机定位至均匀分布在一个圆上。我们首先令 FY02 接收来自另外两台的信号，调整至夹角等于  $60^\circ$ ，再令 FY03 接收来自于另外两台的信号，并沿着来自于 FY01 信号的直线移动至夹角等于  $60^\circ$ ，那么 FY01，FY02，FY03 即定位完毕。



下面使用选择信号源的策略定位剩余的无人机。对于任意给定的数量

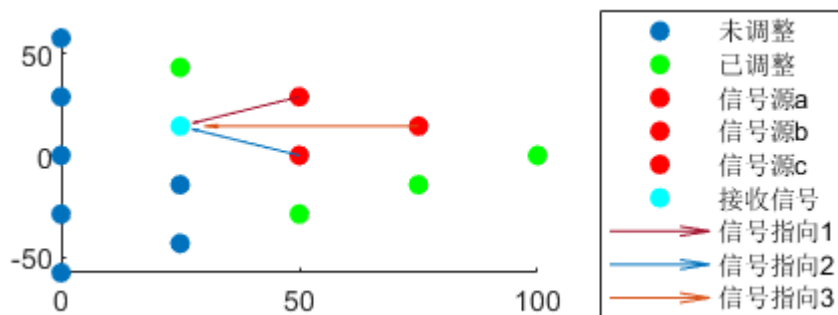
$$N = \frac{L^2 + L}{2} \quad (L \in N^+)$$

我们设计出了，可以计算出这一列上，每一架无人机应该收到来自于哪几架无人机的信号的递推算法程序。表示为  $[a, b, c, d]$  或  $[a, b, c]$ ，每一架飞机第一次接收信号表示为  $[a, b, c, d]$  其中每个数组最后的一项为该无人机的编号，前面的项为需要收到的信号来源的无人机的编号。  $[a, b, c]$  为一架飞机第二次收到调整信号（如有需要），0 表示为空，即没有信号发出。

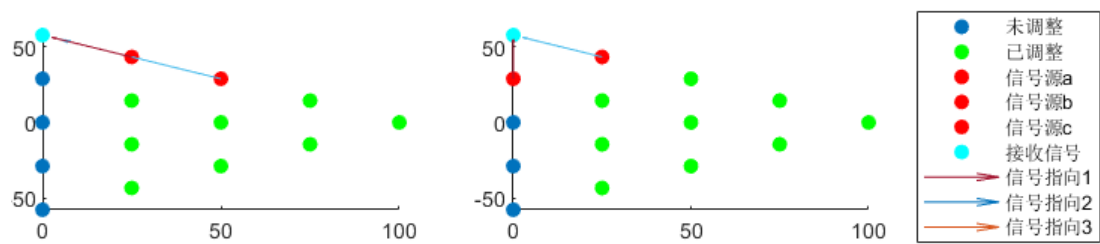
	1	2	3
1	1	[0,0,0,0]	[0,0,0]
2	2	[1,3,2,0]	[0,0,0]
3	3	[1,2,3,0]	[0,0,0]
4	4	[1,2,4,0]	[2,5,4]
5	5	[2,3,1,5]	[0,0,0]
6	6	[1,3,6,0]	[3,5,6]
7	7	[2,4,7,0]	[4,8,7]
8	8	[4,5,2,8]	[0,0,0]
9	9	[5,6,3,9]	[0,0,0]
10	10	[3,6,10,0]	[6,9,10]
11	11	[4,7,11,0]	[7,12,11]
12	12	[7,8,4,12]	[0,0,0]
13	13	[8,9,5,13]	[0,0,0]
14	14	[9,10,6,14]	[0,0,0]
15	15	[6,10,15,0]	[10,14,15]

### 5.3.2 具体方法

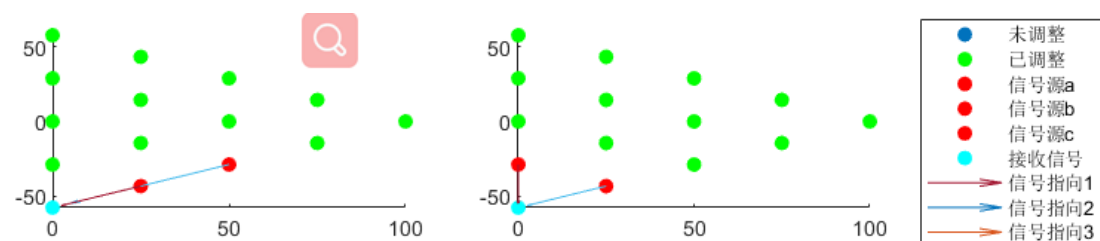
我们以第四列的 FY08 为例，先定位最上方的无人机，需要三架无人机的信号，前一列最上方的无人机 A（如 FY04），前两列最上方的无人机 B（如 FY02），无人机 A 的下一个编号的无人机 C（如 FY05）。调整时需要无人机进行移动，使得两个小夹角相等且等于  $30^\circ$ ，最大的夹角等于  $60^\circ$ ，即调整完毕，继续调整下一位编号的飞机，直到调整到编号为这一列最下方的无人机。



处于三角形上边长上的（题目图中的 FY04, FY07, FY11 等，这里指 FY11）需要来自于前一列，前两列的同处于上边长上的无人机（如 FY07 和 FY04），和自己下一个编号的无人机（如 FY12），总共三架的信号才可以进行定位。通过调整，保证与同边长无人机处于同一直线，再令接收到的另一个角度为  $60^\circ$  即可完成调整。

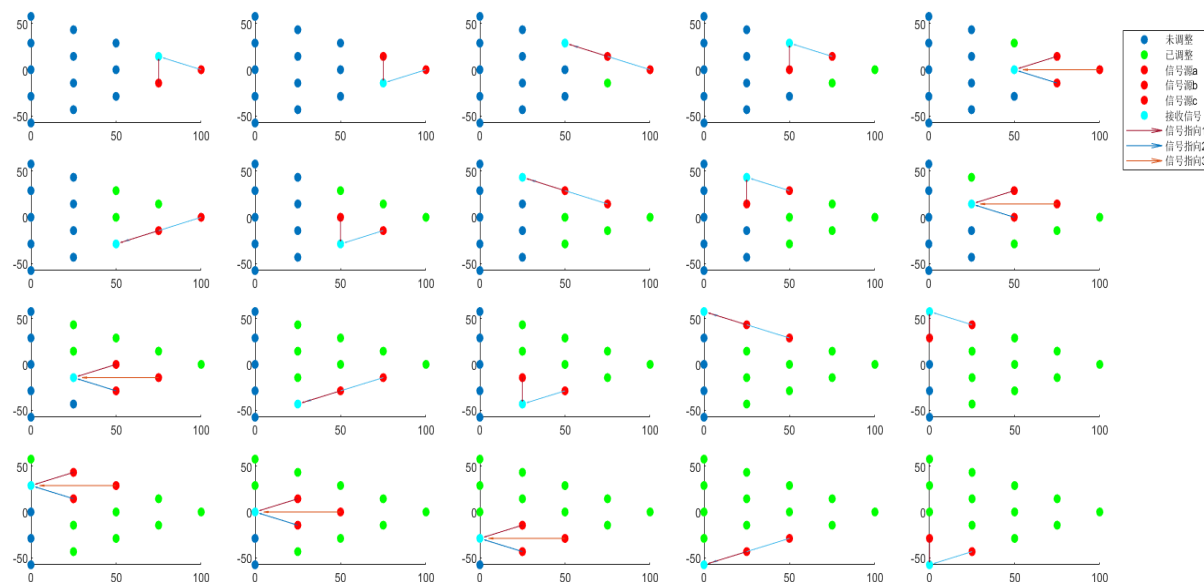


处于三角形下边长上的（题目图中的 FY06, FY10, FY15 等，这里指 FY15）需要来自于，前一列，前两列的同处于下边长上的无人机（如 FY10 和 FY06），和自己上一个编号的无人机（如 FY14），总共三架的信号才可以进行定位。通过调整，保证与同边长无人机处于同一直线，再令接收到的另一个角度为  $60^\circ$ 。



此时这一列的无人机全部定位完毕，并都作为新的信号源去定位下一列的飞机。

完整的过程为：



由此得到的调整次数为 20 次。

由此基于此算法可以得到的结论为：

1. 使用 3 台飞机作为信号源时可以优化调整的次数。
2. 在优先考虑临近飞机的情况下，每一个发射信号过程都可以分解成同样的子过程，从而完成递推的算法。（具体程序参见附录）

## 六、模型的评价和推广

### 6.1 问题一（1）模型的评价

优点：法一运用坐标法便于处理了复杂的几何关系，法二通过三角正弦的关系得出简单的等式。

缺点：无法得出精确解，只能求得近似解。

### 6.2 问题一（2）模型的评价

优点：法一将复杂的求解问题转化成了混合整数非线性规划问题，具有一定的创新型，法二通过数形结合寻求思路，顺利建立了方程组。

缺点：无法得出精确解，只能求得近似解。

### 6.3 问题一（3）模型的评价和推广

优点：方便快捷得出近似解；可以灵活推广至使用别的无人机的情况。

缺点：每一种情况都需要对应求出一组正确的角度；需要人为调整精度系数  $k_1, k_2$ ，迭代次数；存在多余的迭代次数。

改进：可以对最后得出的结果进行约束，进而约束住迭代次数，即无需人为设置迭代次数；可以对权重指数的方程进行细化，使得权重指数更加可信；同理，可以对精度系数的方程进行细化，使得无人机的移动精度得到改进。

### 6.4 问题二模型的评价和推广

优点：给定任意能满足锥形编队队形的无人机数目，都能计算出需要调整的方案，即可以推广至 15 架以上的无人机编队。

缺点：该调整方法只能将整个锥形编队确定在同一平面上。

改进：初始定位时，可通过再增加若干架无人机，使这批无人机保持与同一高度。

## 参考文献

- 【1】 姜启源、谢金星、叶俊 编著，数学模型，北京等教育出版社，2018.
- 【2】 司守奎、孙玺菁 编著，数学建模算法与应用，北京：国防工业出版社，2018.
- 【3】 李昕 编著，MATLAB 数学建模，北京：清华大学出版社，2017.

## 附录

### 附录清单

文件名	文件用途
tanLan.mlx	问题一第 3 小问算法程序
taper.mlx	问题二的算法程序
calCos.m	计算角度的余弦值的函数
calA.m	第一种情形下计算角度的函数
calAngle.m	第二种情形下计算角度的函数
calDis.m	计算距离的函数
圆形调整方案贪婪法.fig	问题一第 3 小问的可视化图像结果
锥形调整方案.fig	问题二的方案的可视化图像结果
算法算出的锥形调整方案表格.jpg	问题二的调整方案 matlab 输出

```
clc,clear
%画出原始图像(有偏移)
r = [0,100,98,112,105,98,112,105,98,112]
th =
deg2rad([0,0,40.10,80.21,119.75,159.86,199.96,240.07,280.17,320.28 ])
%th2 是原始图像
th2 = linspace(0,360,10);
sz = 75;

polarscatter(th,r,sz,'filled')
R = 100;
%polarscatter(deg2rad(th2),R,sz,'filled')

%计算在圆上的正确的 aerfa

real_a = zeros(9,9);

real_a(1,2) = 70;
real_a(1,3) = 50;
real_a(1,4) = 30;
real_a(1,5) = 10;
real_a(1,6) = 10;
```

```
real_a(1,7) = 30;  
real_a(1,8) = 50;  
real_a(1,9) = 70;
```

```
real_a(2,3) = 70;  
real_a(2,4) = 50;  
real_a(2,5) = 30;  
real_a(2,6) = 10;  
real_a(2,7) = 10;  
real_a(2,8) = 30;  
real_a(2,9) = 50;  
real_a(2,1) = 70;
```

```
real_a(3,4) = 70;  
real_a(3,5) = 50;  
real_a(3,6) = 30;  
real_a(3,7) = 10;  
real_a(3,8) = 10;  
real_a(3,9) = 30;  
real_a(3,1) = 50;  
real_a(3,2) = 70;
```

```
real_a(4,5) = 70;  
real_a(4,6) = 50;  
real_a(4,7) = 30;  
real_a(4,8) = 10;  
real_a(4,9) = 10;  
real_a(4,1) = 30;  
real_a(4,2) = 50;  
real_a(4,3) = 70;
```

```
real_a(5,6) = 70;  
real_a(5,7) = 50;  
real_a(5,8) = 30;  
real_a(5,9) = 10;  
real_a(5,1) = 10;  
real_a(5,2) = 30;  
real_a(5,3) = 50;  
real_a(5,4) = 70;
```



```

real_a(6,7) = 70;
real_a(6,8) = 50;
real_a(6,9) = 30;
real_a(6,1) = 10;
real_a(6,2) = 10;
real_a(6,3) = 30;
real_a(6,4) = 50;
real_a(6,5) = 70;

real_a(7,8) = 70;
real_a(7,9) = 50;
real_a(7,1) = 30;
real_a(7,2) = 10;
real_a(7,3) = 10;
real_a(7,4) = 30;
real_a(7,5) = 50;
real_a(7,6) = 70;

real_a(8,9) = 70;
real_a(8,1) = 50;
real_a(8,2) = 30;
real_a(8,3) = 10;
real_a(8,4) = 10;
real_a(8,5) = 30;
real_a(8,6) = 50;
real_a(8,7) = 70;

real_a(9,1) = 70;
real_a(9,2) = 50;
real_a(9,3) = 30;
real_a(9,4) = 10;
real_a(9,5) = 10;
real_a(9,6) = 30;
real_a(9,7) = 50;
real_a(9,8) = 70;
real_a = deg2rad(real_a);
a = zeros(9,9);

```

%为除了原点的每个飞机定义一个信号权重

%目标：将所有飞机均匀分布在半径为 **100** 的圆上

```
th2 = deg2rad(th2)
```

```

for j=1:500
Q = abs(th(3:end) - th2(2:end-1)/th2(2:end-1)) +abs( r(3:end)-R/R);
Q1 = abs (Q-1);
x = find(min(Q1));%这里找出一个除了原点与点 1 之外最好的点
%x+1 即为最好的点的索引
%与此同时需要用原点与第一个点与选中的点来算出其余飞机的  $\alpha$  角

for i = 2:9
    if(i ~= x+1)
        a(i,1) = calA(r(i+1),th(i+1),r(2),th(2));
        a(i,x+1) = calA(r(i+1),th(i+1),r(x+2),th(x+2));
    end
end

%设置 k 的大小
k1 = 5;
k2 = 0.0005;
%调整 d 和  $\theta$ 
for i = 3:10
    if(i~=x+2)
        if(i==7)
            r(i) = r(i) + k1*(max(a(i-1,x:x+1))-max(real_a(i-1,x:x+1)));
        else
            th(i) = th(i) + k2*(min(a(i-1))-min(real_a(i-1)));
            r(i) = r(i) + k1*(max(a(i-1,x:x+1))-min(a(i-1,x:x+1))-
max(real_a(i-1,x:x+1))+min(real_a(i-1,x:x+1)));
            th(i) = th(i) + k2*(min(a(i-1))-min(real_a(i-1)));
        end
    end
end
end
hold on
polarscatter(th,r,sz,'filled')
end

```

taper.mlx 用于解决问题 2

```
clc,clear
```

```
%第四问的标准坐标
```

```
d = 50;
```

```
coordinate= [2*d,0;  
             3/2*d,d/sqrt(3)/2;  
             3/2*d,-d/sqrt(3)/2;...  
             d,d/sqrt(3);  
             d,0;  
             d,-d/sqrt(3);  
             d/2,d*3/sqrt(3)/2;  
             d/2,d/sqrt(3)/2;  
             d/2,-d/sqrt(3)/2;  
             d/2,-d*3/sqrt(3)/2;  
             0,d/sqrt(3)*2;  
             0,d/sqrt(3);  
             0,0;0,-d/sqrt(3);0,-d/sqrt(3)*2];
```

```
out = zeros(15,1);
```

```
%初始条件
```

```
out(1) = 1;
```

```
N = input('请输入符合 $(n^2+n)/2$  的数字（n 为大于 1 的整数）：');
```

```
%通过 N 求解出总列数
```

```
syms x
```

```
l = solve((x^2+x)/2-N==0);
```

```
l = l(2)
```

```
%边上的点的数字 f
```

```
temp = N;
```

```
d(1,1) = N;
```

```
d(1,2) = N - (l - 1);
```

```
for i = 1:l - 1
```

```
    temp = temp - (l-i+1) ;
```

```
    d(l - i,1) =temp
```

```
    d(l-i,2) = d(l - i,1) - (l-i - 1);
```

```
end
```

```
d(1,2) = 1;
```

```
In = {1,[0 0 0 0],[0 0 0]}
```

```
In(2,:) = {2,[1,3,2,0],[0,0,0]}
```

```
In(3,:) = {3,[1,2,3,0],[0,0,0]}
```

```

for i = 3:N
    if (find(i == d(:,2)))
        t = find(i == d(:,2));
        In(i,:) = {i,[d(t-2,2),d(t-1,2) ,i ,0],[d(t-1,2),i+1 i]}
    else if(find(i == d(:,1)))
        if(i>3)
            t = find(i == d(:,1));
            In(i,:) = {i,[d(t-2,1),d(t-1,1) i ,0],[d(t-1,1),i-1 i]}
        end
    else
        %当前列
        currentL = find(i<d(:,1),1)
        In(i,:) = {i ,[i-currentL,i-currentL+1,i-(currentL-1)*2,
i],[0,0,0]};
        end
    end
end

figure_num = 1;
for i = 2: N
    subplot(5,5,figure_num)
    hold on
    scatter(coordinate(:,1),coordinate(:,2),"o","filled")
    t = find(out==1);
    scatter(coordinate(t,1),coordinate(t,2),"o","filled","MarkerFaceColor",
"r")
    Input = In{i,2,1};

    if(Input(4)~=0)
        scatter(coordinate(Input(1),1),coordinate(Input(1),2),"o","filled","Mar
kerFaceColor","r")
        scatter(coordinate(Input(2),1),coordinate(Input(2),2),"o","filled","Mar
kerFaceColor","r")
        scatter(coordinate(Input(3),1),coordinate(Input(3),2),"o","filled","Mar
kerFaceColor","r")
        scatter(coordinate(Input(4),1),coordinate(Input(4),2),"o","filled","Mar
kerFaceColor",[0 1 1])

    quiver
    (coordinate(Input(1),1) ,coordinate(Input(1),2),coordinate(Input(4),1)-
coordinate(Input(1),1),coordinate(Input(4),2)-coordinate(Input(1),2));

```

```

quiver
(coordinate(Input(2),1),coordinate(Input(2),2),coordinate(Input(4),1)-
coordinate(Input(2),1),coordinate(Input(4),2)-coordinate(Input(2),2));
quiver
(coordinate(Input(3),1),coordinate(Input(3),2),coordinate(Input(4),1)-
coordinate(Input(3),1),coordinate(Input(4),2)-coordinate(Input(3),2));
out(Input(1)) = 1;
out(Input(2)) = 1;
out(Input(3)) = 1;
out(Input(4)) = 1;
hold off
figure_num = figure_num+1;
else

scatter(coordinate(Input(1),1),coordinate(Input(1),2),"o","filled","Mar
kerFaceColor","r")
scatter(coordinate(Input(2),1),coordinate(Input(2),2),"o","filled","Mar
kerFaceColor","r")
scatter(coordinate(Input(3),1),coordinate(Input(3),2),"o","filled","Mar
kerFaceColor",[0 1 1])

quiver
(coordinate(Input(1),1),coordinate(Input(1),2),coordinate(Input(3),1)-
coordinate(Input(1),1),coordinate(Input(3),2)-coordinate(Input(1),2));
quiver
(coordinate(Input(2),1),coordinate(Input(2),2),coordinate(Input(3),1)-
coordinate(Input(2),1),coordinate(Input(3),2)-coordinate(Input(2),2));
out(Input(1)) = 1;
out(Input(2)) = 1;
out(Input(3)) = 1;
hold off
figure_num = figure_num+1;
if(i>3)
    subplot(5,5,figure_num)
    Input = In{i,3,1};
    hold on
    scatter(coordinate(:,1),coordinate(:,2),"o","filled")
    t = find(out==1);
    scatter(coordinate(t,1),coordinate(t,2),"o","filled","MarkerFaceColor",
"g")

scatter(coordinate(Input(1),1),coordinate(Input(1),2),"o","filled","Mar
kerFaceColor","r")

```

```

scatter(coordinate(Input(2),1),coordinate(Input(2),2),"o","filled","MarkerFaceColor","r")
scatter(coordinate(Input(3),1),coordinate(Input(3),2),"o","filled","MarkerFaceColor",[0 1 1])

quiver
(coordinate(Input(1),1),coordinate(Input(1),2),coordinate(Input(3),1)-coordinate(Input(1),1),coordinate(Input(3),2)-coordinate(Input(1),2));
quiver
(coordinate(Input(2),1),coordinate(Input(2),2),coordinate(Input(3),1)-coordinate(Input(2),1),coordinate(Input(3),2)-coordinate(Input(2),2));
out(Input(1)) = 1;
out(Input(2)) = 1;
out(Input(3)) = 1;
hold off
figure_num = figure_num+1;
end
end
end

```

calDis.m 用于计算距离

```

function [dis] = calDis(th1,r1,th2,r2)
dis = sqrt(r1^2 + r2^2 - 2*r1*r2*cos(deg2rad(360-th2+th1)));

end

```

calCos.m 用于计算角度余弦

```

function [angleCos] = calCos(r1,r2,r3)
angleCos = (r1*r1 + r2*r2 - r3*r3) / (2*r1*r2);

end

```

calAngle.m 用于计算角度

```

function [angle] = calAngle(r1,r2,r3)
angle = (r1*r1 + r2*r2 - r3*r3) / (2*r1*r2);
angle = acos(angle);
end

```