

无人机遂行编队纯方位无源定位的研究

【摘要】

本文主要对无人机的纯方位无源定位进行研究,通过编队中几架无人机发射信号,其他无人机接收信号,从而确定无人机位置,对发生偏移的无人机进行调整,保持编队整齐,建立不同情况下的定位模型,从而对无人机的位置进行调整。

针对问题一的第一小问,在编队形状为圆形且无人机的架数已知的条件下,还给出了发射信号的无人机架数及其编号,本题采用两种方法都可行,方法一:建立直线方程通过每条直线方程的斜率利用正切定理求出两两间的夹角,从而推导出被接收信号无人机的位置;方法二:通过构建多个三角形,利用正弦定理建立关系式,建立方程组求解得出极径和极角的解析式即是被接收信号无人机的位置。方法一和方法二所推导出的解析式即是定位模型。

针对问题一的第二小问,在发射信号无人机由3架已知变为2架已知,编号分别为FY00和FY01所以它们的位置也是已知,在第一问所求的定位模型基础上,由于已知2架无人机是无法定位的,这是一个角度定值的问题,当被接受无人机只存在略有偏差时,无人机的轨迹的圆心将会确定,我们利用垂径定理及正弦定理计算出无人机轨迹方程,在轨迹方程中求夹角最大的点和夹角最小的点,从而推导出第三架发射信号无人机的编号,再利用问题一的模型进行定位。最终得到本问结论:除了FY00和FY01外,还需要1架无人机发射信号,才能实现有效定位。

针对问题一的第三小问,已知10架无人机的初始极坐标,我们需要对其有误差的位置进行位置调整,由于为了避免外界干扰,无人机应尽可能保持电磁静默,因此在完成位置调整的前提下,我们应该使得总的无人机发射电磁波信号最小,建立优化模型,以贪心思想为调整的原则,让每一次调整无人机位置时使得无人机与其理想位置的差距最小,求得当该调整次数为3次时,无人机与其理想位置的最大差距为 $0.0198m$,当该调整次数为4次时,无人机与其理想位置的最大差距为 $0.000843m$ 。我们选择调整次数为3次,因为其调整量级已经是分米级,误差可以忽略不计,向外发射电磁波信号更少。具体调整方案见5.1.3.7。

针对问题二,与第一问不同的是,本问无人机的编队采用的是锥形编队,无人机的编队队形可以不局限于同一个平面,而是在三维空间中程规律分布,当编队前方出现障碍物时,我们采用旋转矩阵,来实现坐标变换的,利用能够改变向量的方向不改变向量大小的特性,结合问题一的定位模型进行推导锥形编队的定位模型从而实现锥形编队的调整,具体调整方案见5.2.7。

关键词: 贪心思想 最优化模型 几何计算 无源定位 正弦定理 三维旋转矩阵

一、问题重述

1.1 引言

无人机群遂行编队飞行中采用纯方位无源定位，一般无人机接收传输信号的方式为定位站发射电磁波进行转递，接收信号后进行调整，而在无人机在实际飞行中会受到外界电磁波的干扰，而纯方位无源定位中无源也就是没有定位站电磁波信号来进行定位，从而达到编队的稳定性。

采用纯方位无源定位的无人机编队需要在队中固定发射信号的无人机，且发射信号的无人机架数需要2架或以上，由于无人机发射信号被动接收信号的无人机只能受到方位信息，一架无人机发射信号并不能使被动接收无人机得到准确的方位信息即无法形成夹角，所以2架或以上无人机发射信号能与被动接收信号无人机形成夹角，通过判断夹角的大小从而确定位置。

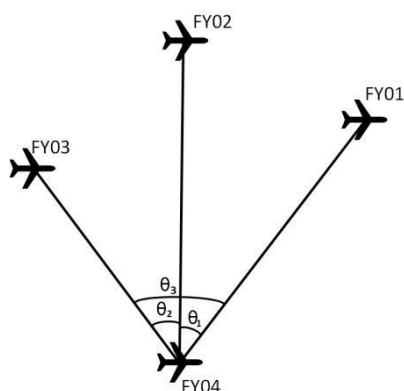


图 1：无人机接收到的方向信息示意图

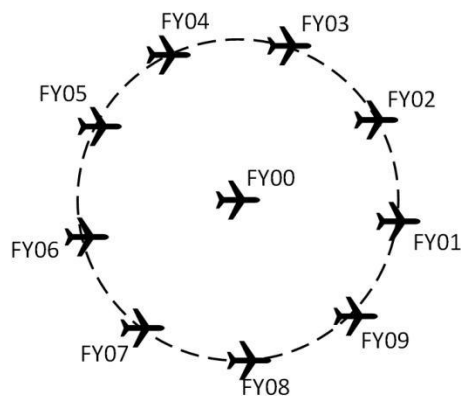


图 2：圆形无人机编队示意图

1.2 问题提出

根据题目的信息，结合已知的不同编队类型的无人机位置信息，建立数学模型研究以下问题：

1. 无人机编队一共10架，队形呈圆形编队，10架无人机的编号已知，除编号为FY00的无人机位于圆心处，其他9架无人机均匀分布在圆周上，同时10架无人机的飞行高度一致，基于上述给出的基本要求，研究以下3个小问。

（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。当发射信号的无人机位置无偏差且编号已知时，建立被动接收信号无人机的定位模型。

（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。若发射信号的无人机位置无偏差，除FY00和FY01外还需要几架无人机发射信号，才能实现无人机的有效定位？

(3) 根据编队要求，一架无人机位于圆心也就是编号为FY00的无人机，其余无人机均匀分布在半径为 100 m 的圆周上，且初始时刻无人机位置会发生偏差，给出合理的无人机位置调整方案；通过表一给出的无人机初始位置极坐标，发射信号的无人机最多为4架，且编号FY00无人机每次都作为发射信号无人机，其余无人机被动接受信号，对无人机的位置进行调整，使9架无人机均匀分布在圆周上，给出具体的调整方案。

2. 在实际飞行中，无人机集群也可以是其他编队队形，例如锥形编队队形。仍考虑存方位无源定位，设计无人机的调整方案

二、问题分析

本题主要是对无人机集群在遂行编队飞行时确定位置并进行调整的方案进行研究。针对问题1的第1小问，需给出发射信号无人机无偏差且编号已知时，被动接收信号无人机的定位模型；针对问题1的第2小问，需给出除FY00和FY01外，需要发射信号的无人机架数；针对问题1的第3小问，需分别给初始时刻、后续无人机的调整方案；针对问题2，当编队队形不再是圆时，设计并给出无人机位置调整方案。结合以上四个具体问题，分别做如下分析：

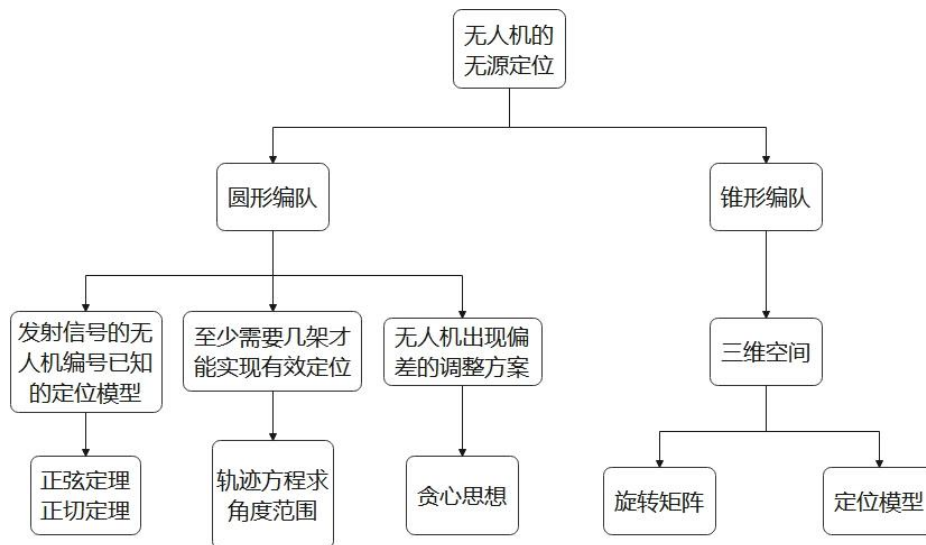


图3：无人机思维导图

2.1 问题 1 分析

在第一问中，编队由10架无人机组成，且为圆形编队，其中1架位于圆心，其它9架均匀分布在圆周上，围绕该条件分别对3个小问进行以下具体分析：

(1) 第一小问分析

在第一问的第一小问中，我们采用2种方法对本题进行求解，由题目已知发射信号的无人机的架数及其编号，通过建立坐标系，得到对应无人机的坐标信息，

由两点式分别列出两架发射信号的无人机与接受信号无人机的直线方程，即可推导出两线的斜率，得到无人机与两架发射信号无人机连线之间的夹角，建立两个方程间关系，联立方程组进行求解，确定被动接受信号的无人机坐标。

第二种方法是采用极坐标的方式进行计算与第一种直接采用坐标进行计算不同，通过两两连接起4个点，建立三角形后利用正弦定理写出各各角度与长度间的关系式，最后转换为关于角度关系式，从而我们根据角度关系，就可以确定被动接收信号无人机的坐标。

(2) 第二小问分析

在第一问的第二小问中，由题目可知的有2架发射信号的无人机FY00和FY01，由此可得2架无人机的位置，当只使用两架无人机的位置对无人机 A_3 进行定位时，是无法定位的因为这是一个角度定值的问题，2架无人机和被动接收信号无人机的连线形成的角度固定的，所以无人机的位置是一段轨迹，可以采用垂径定理以及正弦定理求出新增无人机与被动接收无人机所形成的角度关系，从而我们可以确定增加的无人机编号。

(3) 第三小问分析

在第一问的第三小问中，已知10架无人机的初始极坐标，可得其中两架无人机的位置准确无误的，其余8架无人机的位置是有偏差的，我们需要对其进行位置调整，由于为了避免外界干扰，无人机应尽可能保持电磁静默，少向外发射电磁波信号，因此在完成位置调整的前提下，我们应该使得总的无人机发射电磁波信号次数最小，建立优化模型。

2. 2问题2分析

在第二问中，无人机编队不在是圆形编队，可以任意编队，由于在无人机实际飞行过程中，无人机的编队队形可以不局限于同一个平面，而是在三维空间中程规律分布，同时需要考虑纯方位无源定位的情形，设计无人机位置调整方案，而对于本问，需要给出无人机锥形编队队形下的无人机位置的确定以及对应调整方案。

三、符号说明

符号	符号说明
f_{ij}	第 i 次调度时第 j 架无人机是否被选择
r_k	第 k 架无人机计算的极半径

x_{ij}	第 <i>i</i> 次调度时第 <i>j</i> 架无人机的 <i>x</i> 坐标
y_{ij}	第 <i>i</i> 次调度时第 <i>j</i> 架无人机的 <i>y</i> 坐标
β_{ij}	第 <i>i</i> 次调度时第 <i>j</i> 架无人机的极角
r_{ij}	第 <i>i</i> 次调度时第 <i>j</i> 架无人机的极径
θ_j	第 <i>j</i> 架无人机的在理想位置的极角
E_{ij}	第 <i>i</i> 次调整后第 <i>j</i> 架无人机与理想的距离
g_{ij}	第 <i>i</i> 次调整后第 <i>j</i> 架无人机的现实坐标

四、模型假设

- 1.假设发射信号的无人机在飞行过程中不会受到风向、风速的影响。
- 2.假设无人机飞行过程中不会受到其他飞行物的干扰。
- 3.假设无人机在发射和接收信号时信号稳定不受干扰。
- 4.假设无人机发射信号不会产生丢失。
- 5.假设无人机都知道自身编号和相对位置。
- 6.假设锥形编队无人机在飞行过程中会遇到障碍物。

五、模型的建立与求解

针对本题不同情况下的无人机接受发射信号，通过给定的发射信号的无人机编号与个数，需给出在圆形编队和锥形编队下发生偏差的无人机的调整方案。建立数学模型，设计相应算法，得出最优的具体调整方案。

5.1 问题 1: 无人机圆形编队的定位和调整

在本问题中，我们需要解决三个子问题。这三个子问题都基于一个圆形编队，其中有一架无人机位于圆心，其他九架无人机均匀分布在圆周上，并且这十架无人机的飞行高度相同。在第一个子问题中，我们采用了两种不同的方法来进行定位。一种是基于正弦定理的无人机定位，另一种是基于正切定理的无人机定位。通过这两种方法，我们可以建立无人机被动接收信号模型；在第二问中基于第一问建立的定位模型，通过任意增加一个发射信号的无人机，在第二小问中通过计算未知编号发射信号无人机与被动接收信号无人机的夹角范围，从而判断新增无人机的编号。

5.1.1 无人机圆形编队定位分析

在第一问中，根据给得的发射信号无人机FY00和另外2架由于是已知条件所以可以进行合理假设另外2架无人机的位置及其编号，我们提供了两种推导方法，一种是基于正弦定理的无人机定位，另一种是基于正切定理的无人机定位。

(1) 基于正弦定理的无人机定位

通过已知的3架无人机的位置和1架被动接收无人机的位置，我们可以建立它们之间的关系。通过连接这4个点，我们可以得到多个不同的三角形。然后，我们可以利用正弦定理建立3个已知点与未知点之间的关系。这样就可以建立2个关于正弦定理的方程。然而，我们有3个未知量，需要3个方程才能求解。为了解决这个问题，我们还需要考虑三角形内角和的关系，最终将推导出的表达式中所有未知量表示出来。这样推导出的式子就是定位模型。

(2) 基于正切定理的无人机定位

通过建立直角坐标系并将点位标注在坐标中，可以利用发射信号点和接收点之间的两点式建立直线方程，从而得到关于接收点的方程。结合斜率的公式，将方程代入，我们可以求得接收信号无人机与两架发射信号无人机之间的夹角。利用另外3架发射信号无人机，我们可以建立3个直线方程，进而得到2条夹角的公式。最后，将这些方程联立起来形成一个方程组，我们就可以求解被动接收信号无人机的坐标。这个方程组即为本问题的定位模型。

5.1.1.1方法一：基于正弦定理的无人机定位的推导

由题目给出的信息我们可以了解到，无人机飞行时采用纯方位无源定位确定位置，在定位时我们无人机接收到某编号发射来的电磁波信号，接收信号的无人机从电磁波信号中可读取的信息有：信号的发送者 A_n ，无人机夹角 θ 信息。在编队中无人机间的相对位置是保持不变的即：各无人机存储了无人机 A_n 的直角坐标系坐标 (x_n, y_n) ，以及极坐标系坐标 (r_n, θ_n) 。直角坐标与极坐标的关系如下：

$$\begin{cases} x_n = r_n \cos(\theta_n) \\ y_n = r_n \sin(\theta_n) \end{cases}$$

在圆形编队中，各无人机均匀分布在半径为 r 的圆上，其中无人机 A_0 为直角坐标系中的原点、极坐标系中的极点，半径为 100m 的圆形编队各无人机坐标如下表所示：

表 1：无人机坐标转换表

无人机编号	直角坐标	极坐标
-------	------	-----

0	(0, 0)	(0, 0)
1	(100, 0)	(100, 0)
2	(76.60, 64.28)	(100, 0.70)
3	(17.36, 98.48)	(100, 1.40)
4	(-50, 86.60)	(100, 2.09)
5	(-93.97, 34.20)	(100, 2.80)
6	(-93.97, -34.20)	(100, -3.50)
7	(-50, -86.60)	(100, 4.19)
8	(17.36, -98.48)	(100, -98.48)
9	(76.60, -64.29)	(100, 5.59)

现已知 A_0 无人机和编队中另 2 架无人机发射信号，且发射信号的无人机位置无偏差和编号已知，而接收信号的无人机位置有偏差。为调整接收信号的无人机位置，我们需要建立极坐标系。三架发射信号的无人机如图所示：

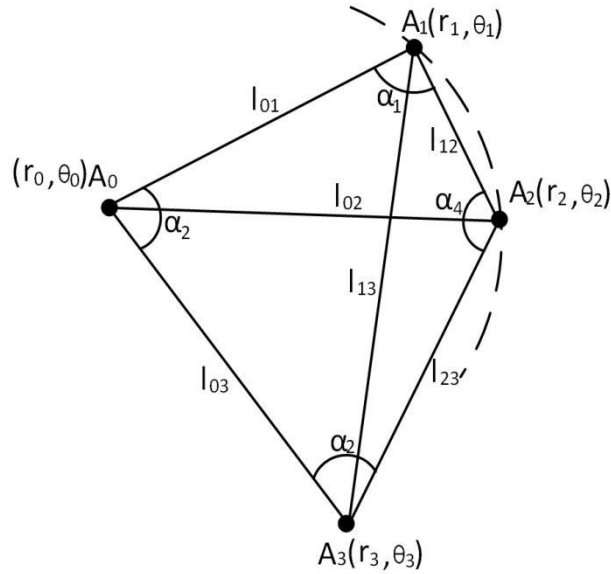


图 4：推导示意图

假设发射信号的三架无人机为 A_0 ， A_1 ， A_2 ，结合上表我们可以将三架无人机对应的极坐标做如下所示：

$$A_n = (r_n, \theta_n) \quad n = 0, 1, 2$$

对于被动接收信号的无人机设为 A_3 ，其极坐标是未知。为了计算出无人机 A_3 的坐标我们采用正弦定理进行推导，首先我们根据正弦定理结合上图所构建的点与点之间的推导示意图分别列出 $\triangle A_0 A_1 A_3$ 、 $\triangle A_1 A_2 A_3$ 、 $\triangle A_0 A_2 A_3$ 的边角关系式，具体如下：

$$\begin{aligned}
k_1 &= \frac{l_{01}}{\sin(\angle A_0 A_3 A_1)} = \frac{l_{13}}{\sin(\angle A_1 A_0 A_3)} = \frac{l_{03}}{\sin(\angle A_0 A_1 A_3)} \\
k_2 &= \frac{l_{12}}{\sin(\angle A_1 A_3 A_2)} = \frac{l_{13}}{\sin(\angle A_1 A_2 A_3)} = \frac{l_{23}}{\sin(\angle A_2 A_1 A_3)} \\
k_3 &= \frac{l_{02}}{\sin(\angle A_0 A_3 A_2)} = \frac{l_{03}}{\sin(\angle A_0 A_2 A_3)} = \frac{l_{23}}{\sin(\angle A_2 A_0 A_3)}
\end{aligned}$$

其中, l_{ij} 为点 A_i 到点 A_j 的距离。

根据上图在, 四边形 $A_0 A_1 A_2 A_3$ 中, 我们可以知道内角和为 360 度, 而 $\angle A_0 A_1 A_2$ 、 $\angle A_0 A_3 A_2$ 也是已知的, 故我们可以根据角度关系推导出 $\angle A_1 A_0 A_3$ 与 $\angle A_1 A_2 A_3$ 的关系式, 具体式子如下:

$$2\pi - \alpha_{012} - \alpha_{032} = \alpha_{103} + \alpha_{123}$$

其中, α_{012} 为 $\angle A_0 A_1 A_2$, α_{032} 为 $\angle A_0 A_3 A_2$, α_{103} 为 $\angle A_1 A_0 A_3$, α_{123} 为 $\angle A_1 A_2 A_3$ 。

联立上述三个方程即可求出边 $A_1 A_3$ 和 $\angle A_1 A_0 A_3$ 以及 $\angle A_1 A_2 A_3$, 进而求出无人机 A_3 的极坐标, 具体公式推导如下:

$$\begin{cases} \frac{l_{13}}{\sin(\alpha_{103})} = k_1 \\ \frac{l_{13}}{\sin(\alpha_{123})} = k_2 \\ \alpha_{123} = 2\pi - \alpha_{012} - \alpha_{032} - \alpha_{103} \end{cases} \Rightarrow \begin{cases} \tan(\alpha_{103}) = \frac{-k_2 \sin(\alpha_{012} + \alpha_{032})}{k_1 + k_2 \cos(\alpha_{012} + \alpha_{032})} \\ \alpha_{123} = 2\pi - \alpha_{012} - \alpha_{032} - \alpha_{103} \\ l_{13} = k_1 \sin(\alpha_{103}) \end{cases}$$

在 $\triangle A_0 A_1 A_3$ 中已知 $\angle A_0 A_3 A_1$ 和 $\angle A_1 A_0 A_3$ 、参数 k_1 , 即将 $\angle A_0 A_1 A_3$ 和 l_{03} 用式子表达出来具体如下:

$$\begin{cases} \alpha_{031} = \pi - \alpha_{103} - \alpha_{031} \\ l_{03} = k_1 \sin(\alpha_{031}) \end{cases}$$

其中, α_{031} 为 $\angle A_0 A_3 A_1$, α_{013} 为 $\angle A_0 A_1 A_3$, 而无人机 A_3 的角度 θ_3 易得, 具体如下:

$$|\theta_3 - \theta_1| = \alpha_{103} \Rightarrow \theta_3 = \{\theta_1 + \alpha_{103}, \theta_1 - \alpha_{103}\}$$

由于, θ_3 角有两种取值, 为得到真正我们所需要的 θ_3 角, 有以下联立:

$$\begin{cases} k_2 = \frac{l_{23}}{\sin(\alpha_{213})} \\ k_3 = \frac{l_{23}}{\sin(\alpha_{203})} \\ \alpha_{213} = \pi - \alpha_{132} - \alpha_{123} \end{cases} \Rightarrow \alpha_{203} = \arcsin\left(\frac{k_2}{k_3} \sin(\alpha_{132} + \alpha_{123})\right)$$

$$|\theta_3 - \theta_2| = \alpha_{203} \quad \Rightarrow \quad \theta_3 = \{\theta_2 + \alpha_{203}, \theta_2 - \alpha_{203}\}$$

其中, α_{213} 为 $\angle A_2 A_1 A_3$, α_{203} 为 $\angle A_2 A_0 A_3$, θ_3 角取值为上述两集合的交集。

$$\theta_3 = \{\theta_2 + \alpha_8, \theta_2 - \alpha_8\} \cap \{\theta_1 + \alpha_3, \theta_1 - \alpha_3\}$$

通过上诉的推导过程, 我们进行整理和后可以得到最后的推导式为:

$$\tan \theta_3 = \frac{\sin \alpha_{031} \sin \alpha_{032} (\cos \theta_1 - \cos \theta_2) \mp \sin \alpha_{032} \cos \alpha_{031} \sin \theta_1 \pm \sin \alpha_{031} \cos \alpha_{032} \sin \theta_2}{\sin \alpha_{031} \sin \alpha_{032} (\sin \theta_2 - \sin \theta_1) \mp \sin \alpha_{032} \cos \alpha_{031} \cos \theta_1 \pm \sin \alpha_{031} \cos \alpha_{032} \cos \theta_2}$$

$$r = \frac{L_{01}}{\sin(\alpha_{031})} \sin(\alpha_{031} + \alpha_{103})$$

根据下图建立的极坐标, 对上述的公式中 $\sin \theta_1$ 和 $\sin \theta_2$ 的正负进行讨论, 极坐标图如下所示:

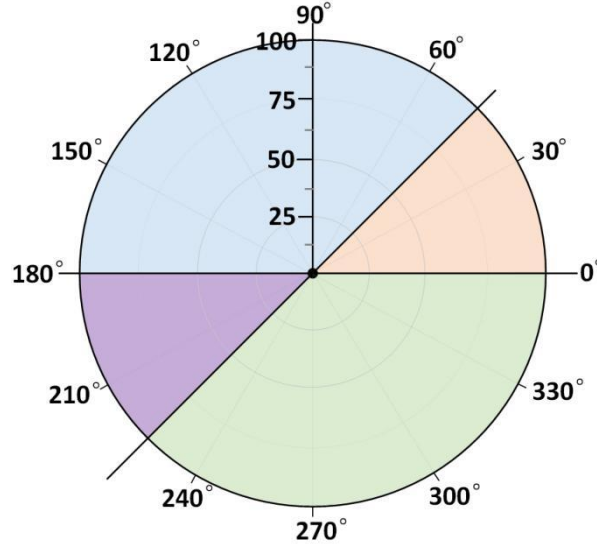


图5: 极坐标示意图

首先由 0° 到 180° 引出一条直线, 在这条直线的上方即是 $\sin \theta_1$ 为负, 也就是橙色区域与蓝色区域, 在线的下方为正, 也就是绿色和紫色区域。同理可得, 在图上引出另一条线, 在这条直线的上方即是 $\sin \theta_2$ 为正, 也就是蓝色和紫色区域, 在线的下方为负, 也就是绿色和橙色区域。

5.1.1.2 方法二: 基于正切定理的无人机定位的推导

由于无人机编队由 10 架无人机组成, 形成圆形编队, 其中 9 架无人机 (编号 FY01~FY09) 均匀分布在某一圆周上, 另 1 架无人机 (编号 FY00) 位于圆心 (见图 2)。无人机基于自身感知的高度信息, 均保持在同一个高度上飞行, 因此, 我们以编号 FY00 的无人机为圆心, 建立直角坐标系。

无人机是以FY00为圆心，并且编队中的其余无人机都均匀分布在某一圆周上，因此，它们应在编队中的位置是固定不变的，当位置略有偏差的无人机接收信号时，接收到的方向信息如下：

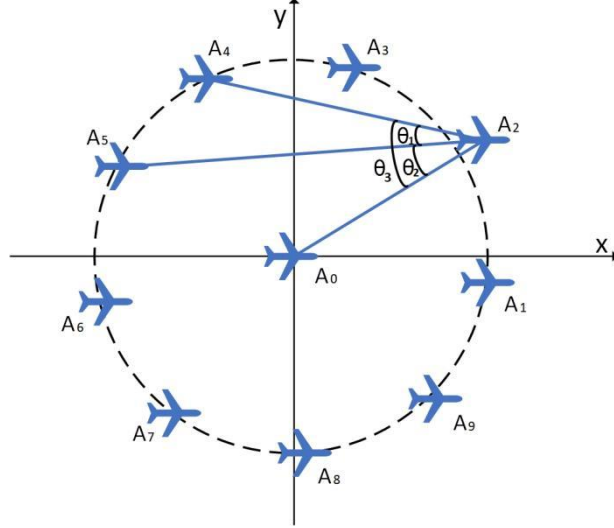


图6：无人机定位示意图

当无人机是以位于圆心的无人机以及编队中的另外两架无人机发射信号，由于发射信号的无人机位置是无偏差并且编号已知的，我们是知道它们的具体位置，圆心（FY00）的无人机 A_0 坐标是为 $(0,0)$ ，其余两架已知编号（位置已知）的无人机的位置我们分别将其设为 $A_1(x_1, y_1)$ ， $A_2(x_2, y_2)$ ，最后我们将位置略有偏差（位置未知）的无人机位置设为 (x_3, y_3) 。

我们需要解得位置略有偏差的无人机的具体位置，因此，我们构建发射信号的无人机与被动接收信号无人机的两点式直线方程, 由上图我们两两创建直线方程，可以创建3个直线方程，具体如下所示：

$$\frac{x-x_3}{-x_3} = \frac{y-y_3}{-y_3}, \quad \frac{x-x_3}{x_1-x_3} = \frac{y-y_3}{y_1-y_3}, \quad \frac{x-x_3}{x_2-x_3} = \frac{y-y_3}{y_2-y_3}$$

我们需要运用两直线的斜率与它们的夹角之间的关系解得被动接收信息的无人机 A_4 的位置，因此，我们分别求出三架发送信号的无人机 A_0 A_1 A_2 与被动接收信息的无人机 A_4 的直线方程所对应的斜率 k_0 k_1 k_2 ，求得的具体表达式如下：

$$k_0 = \frac{y_3-0}{x_3-0}, \quad k_1 = \frac{y_3-y_1}{x_3-x_1}, \quad k_2 = \frac{y_3-y_2}{x_3-x_2} \quad (1)$$

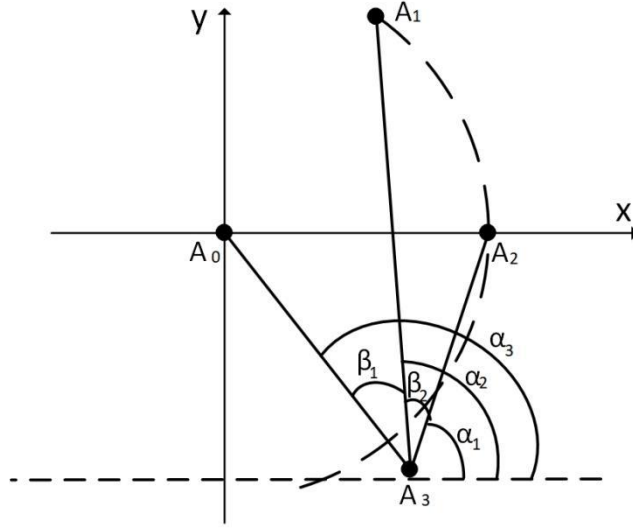


图7：推导示意图

接着运用两直线的夹角与斜率的之间关系，由于我们有2个夹角，所以需要写出2个夹角与斜率间的关系式，具体表达式如下：

$$\begin{cases} \tan \theta_1 = \frac{k_1 - k_0}{1 - k_1 k_0} \\ \tan \theta_2 = \frac{k_2 - k_0}{1 - k_2 k_0} \\ \tan \theta_3 = \frac{k_2 - k_1}{1 - k_2 k_1} \end{cases} \quad (2)$$

将无人机 A_0 A_1 A_2 与被动接收信息的无人机 A_4 的直线方程所对应的斜率 k_0 k_1 k_2 推导式也就是推导式①代入两直线的夹角与斜率的之间关系式也就是公式②进行展开并化简得到如下式子：

$$\tan \theta_1 = \frac{x_1 y_2 - y_1 x_2}{-x_1 x_2 + x_1^2 - y_1 y_2 + y_1^2} \quad (3)$$

同理可得另一个夹角为：

$$\tan \theta_2 = \frac{x_1 y_4 - y_1 x_4}{-x_1 x_4 + x_1^2 - y_1 y_4 + y_1^2} \quad (4)$$

最后我们将推导式③和推导④写成关于无人机 A_4 人坐标的表达式，就是本问的定位模型，具体表达式如下：

$$\begin{cases} x_i = \frac{y_1x_2 - px_2 + px_1 - y_2x_1 - (p^2 - py_1 - py_2 + y_1y_2 - x_1x_2) \tan(\theta_3)}{-\tan(\theta_3) + y_1 \tan(\theta_3) + y_2 \tan(\theta_3) - x_2 + x_1} \\ y_i = \frac{x_1y_2 - qy_2 + qy_1 - x_2y_1 - (q^2 - qx_1 - qx_2 + x_1x_2 - y_1y_2) \tan(\theta_3)}{-\tan(\theta_3) + y_1 \tan(\theta_3) + y_2 \tan(\theta_3) - x_2 + x_1} \\ p = \frac{Cx_2 - Cy_2 + Dy_1 - Dy_1}{C - D} \\ q = \frac{Ay_2 - Ay_1 + Bx_1 - Bx_1}{A - B} \\ A = (x_3^2 - x_1x_3) \cdot \tan(\theta_2) \tan(\theta_1) - y_1x_3 \tan(\theta_2) \\ B = (x_3^2 - x_1x_2) \cdot \tan(\theta_1) \tan(\theta_2) - y_2x_3 \tan(\theta_2) \\ C = (y_3^2 - y_1y_3) \cdot \tan(\theta_2) \tan(\theta_1) - x_1y_3 \tan(\theta_2) \\ D = (y_3^2 - y_1y_2) \cdot \tan(\theta_1) \tan(\theta_2) - x_2y_3 \tan(\theta_2) \end{cases}$$

5.1.1.3 模型的定位效果分析

综上所述，对基于正弦定理推导出的解析式，利用Matlab编程求解定位模型的误差，通过给出测试样例后，利用我们的模型对被接收信号无人机的位置进行定位，根据我们模型的定位结果和给的测试样例的结果进行比对。

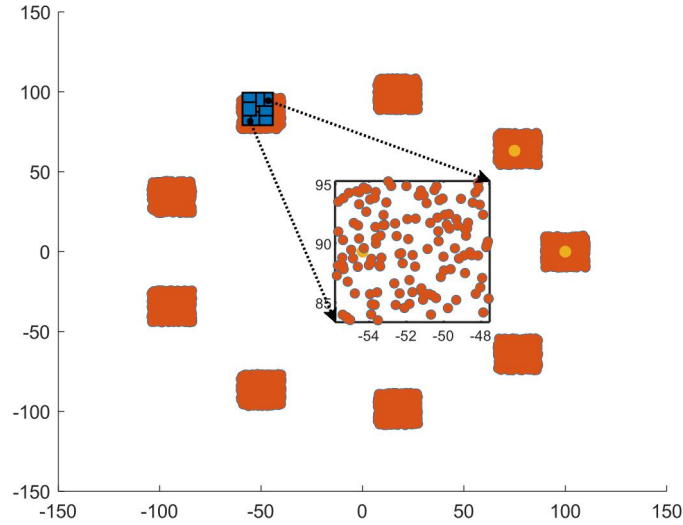


图8：测试样例与定位效果示意图

比对的方法我们采用的是两点距离公式，因为我们求出的是被接受无人机的坐标，所以采用两点距离公式能够求出测试样例和我们所求结果的距离，从而可以判断出我们计算出的结果与测试样例的距离，也就能确定我们模型计算的准确情况。

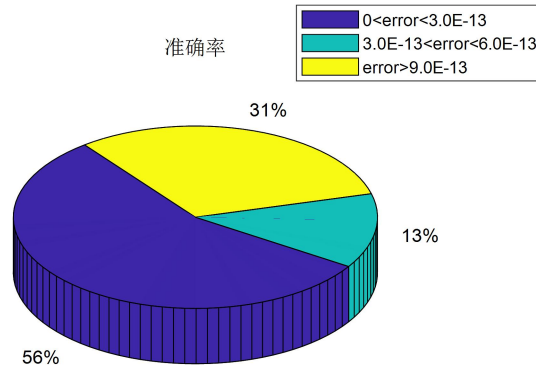


图9: 准确率饼图

因为我们采用5000组测试用例数据量比较庞大，所以采用饼图的方式来呈现我们的准确率，由上图可以清楚看到，我们的模型准确率非常高，误差几乎为0，我们的大部分计算出的数据都居于 $0 < \text{error} < 6 \times 10^{-13}$ 之间，由此可以判断出，我们推导出的定位模型效果非常好。

5.1.2 确定增加的无人机架数

在第一小问中，我们已经求出被动接收无人机的定位模型，在本问中由于只是已知 A_0 ， A_1 两架无人机的编号也就是这两架无人机的位置已知，第3架以及第 i 架无人机发射信号的无人机位置是未知的，计算最少还需要多少架无人机才能对于无人机进行有效定位。由此我们可以得到，只是2架无人机已知是无法对被动接收信号的无人机进行定位的，我们考虑采用垂径定理以及正弦定理来进行本问推导。

5.1.2.1 增加无人机架数模型的推导

当只使用 A_0 ， A_1 两架无人机的位置对无人机 A_3 进行定位时，是无法定位的因为这是一个角度定值的问题，2架无人机和被动接收信号无人机的连线形成的角度固定的，所以无人机 A_3 的位置是一段轨迹，示意图如下所示：

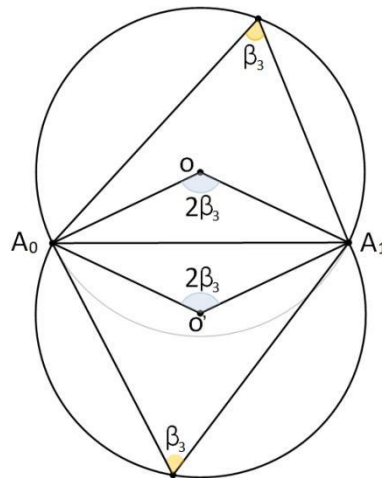


图10: 点位轨迹示意图

但是被动接收信号的无人机是明白自己的编号，当只存在略有偏差时，无人机的轨迹的圆心将会确定，我们利用垂径定理以及正弦定理计算出无人机轨迹方程如下：

①当接收信号的无人机编号为 $A_2 - A_5$ 时：

$$\left(x - \frac{R}{2}\right)^2 + \left(y - \frac{R}{2\cos(\beta_3)}\right)^2 = \left(\frac{R}{2\tan(\beta_3)}\right)^2 \quad y \leq 0$$

②当接收信号的无人机编号为 $A_6 - A_9$ 时：

$$\left(x - \frac{R}{2}\right)^2 + \left(y + \frac{R}{2\cos(\beta_3)}\right)^2 = \left(\frac{R}{2\tan(\beta_3)}\right)^2 \quad y \geq 0$$

例如当接收信号的无人机为 A_3 时，无人机轨迹的圆心将为点 o ，并且真正的轨迹与无人机应在的位置略有偏差时，公式如下：

$$\begin{cases} \left(x - \frac{R}{2}\right)^2 + \left(y - \frac{R}{2\cos(\beta_3)}\right)^2 = \left(\frac{R}{2\tan(\beta_3)}\right)^2 & x_i - 15 \leq x \leq x_i + 15, y_i - 15 \leq y \leq y_i + 15, i = 2 \dots 5 \\ \left(x - \frac{R}{2}\right)^2 + \left(y + \frac{R}{2\cos(\beta_3)}\right)^2 = \left(\frac{R}{2\tan(\beta_3)}\right)^2 & x_i - 15 \leq x \leq x_i + 15, y_i - 15 \leq y \leq y_i + 15, i = 6 \dots 9 \end{cases}$$

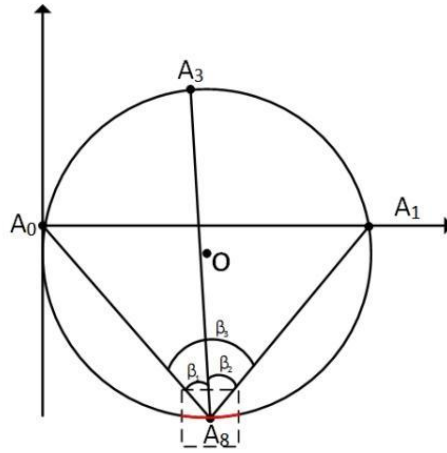


图11：角度范围示意图

当求得接收信号的无人机可能的轨迹时，我们再次增加一架无人机进行无源定位。在圆形编队中， $A_0 - A_9$ 无人机均匀分布在圆周上。 A_0 和 A_1 两架无人机已确定为发送信号的无人机，我们遍历 $A_2 - A_9$ 中的另外7架无人机，只要它们在这段轨迹中接收到符合接收信号的方向信息，即可确定接收信号的无人机的编号。

然而，由于轨迹中设置的步长可能会导致漏掉符合条件的点。因此，我们在轨迹中寻找与 A_0 和 A_1 夹角最大的点和夹角最小的点，即轨迹的最左端和最右端。

一旦 β_1 在夹角范围之内，即可确定发送无人机的编号为 k 。我们将接收信号的无人机假设为 A_8 。具体公式如下：

$$\begin{cases} \beta_1 \in f_k[\min \angle A_0 A_8 A_k, \max \angle A_0 A_8 A_k] \\ f_k \in \{0,1\} \end{cases}$$

β_1 是表示为接收信号的无人机 A_8 与发射信号无人机 $A_0 A_k$ 之间实际的夹角，而由于接收信号的无人机 A_8 与发射信号无人机 $A_0 A_1$ 之间的夹角是 β_3 ，这在轨迹上 β_3 都是固定不变，当 $\angle A_0 A_8 A_k$ 确定为 β_1 了， $\angle A_1 A_8 A_k$ 也就确定为 β_2 了，所有方向信息就都会准确。

将其发射信号的无人机的编号求出为第 k 个无人机，我们就可以知道第 k 架无人机的极坐标就为 $(R, \frac{2k\pi}{9})$ ，因此我们可以得出三架发射信号的无人机的准确坐标，代入到我们在第一小问推导的模型中，具体模型如下：

$$\beta_k = \arctan \frac{\sin \alpha_{0k1} \sin \alpha_{0k8} (\cos \beta_1 - \cos \beta_2) \mp \sin \alpha_{0k8} \cos \alpha_{0k1} \sin \beta_1 \pm \sin \alpha_{0k1} \cos \alpha_{0k8} \sin \beta_2}{\sin \alpha_{0k1} \sin \alpha_{0k8} (\sin \beta_2 - \cos \beta_1) \mp \sin \alpha_{0k8} \cos \alpha_{0k1} \cos \beta_1 \pm \sin \alpha_{0k1} \cos \alpha_{0k8} \cos \beta_2}$$

$$r = \frac{L_{01}}{\sin(\alpha_{0k1})} \sin(\alpha_{0k1} + \alpha_{10k})$$

5.1.2.2 具体结果

综上所述，结合上节建立的模型，利用matlab求解出当接收信号的无人机为 A_8 时，新增发射信号的无人机分别与 $A_0 A_8$ ， $A_1 A_8$ 所对应的角度应在的范围。如下表：

表2：新增发射信号无人机角度范围

新增无人机	β_1 最小值	β_1 最大值	β_2 最小值	β_2 最大值
A_2	23.86°	37.86°	16.82°	22.89°
A_3	5.24°	16.48°	34.84°	44.84°
A_4	4.65°	15.49°	53.16°	66.62°
A_5	24.63°	37.68°	71.34°	88.77°
A_6	41.97°	61.29°	88.69°	112.38°
A_7	56.01°	91.79°	102.72°	142.88°
A_9	87.66°	87.73°	11.16°	27.67°

根据上表，我们给出了除了 $A_0 A_1$ 为发射信号无人机、和假设 A_8 为被动接收

信号无人机，其余编号无人机与被接受无人机 A_8 为形成的夹角范围，如果新增的无人机处于上述角度的范围，那么我们就可以确定增加的无人机编号为多少。

我们以上表内容举一个例子，如选择发射信号的无人机为 $A_0 A_8 A_i$ ，当我们用上诉模型将 $A_8 A_i$ 的夹角 β_1 计算出来的角度假如为 24.86° ，此时通过查表可以看到该角度处于 $23.86^\circ - 37.86^\circ$ 之间，说明增的无人机编号为 A_2 。同理可得，我们也可以选择 $A_1 A_8 A_i$ ，将 $A_8 A_i$ 的夹角 β_2 计算出来，查表可得增加的无人机编号。所以本题我们除了FY00和FY01外，还需要1架无人机发射信号，才能实现有效定位。

5.1.3 调整无人机位置使其均匀分布于圆周上

在本问中，给出了10架无人机的初始位置极坐标，通过转换为坐标得到具体位置，结合第二问的结果得到至少需要三只发射信号无人机才可确定方位，所以选择编号为FY00和另外随机2架无人机作为发射信号，而被动接收信号的无人机只能根据发射信号无人机的方位进行调整，于是由第一小问的定位模型去求得接收信号无人机此时认为准确的坐标。

但是发射信号的无人机本身就存在一定偏差，所以被接收信号所接收到的方向信息也是存在偏差，而我们需要将偏差的点往理想位置进行移动，它们之间的差值就是偏差值。建立最优化模型，利用 *Matlab* 编程求解。

5.1.3.1 圆形编队位置预设

在本小题中，无人机的编队设置为圆形编队，其中一架无人机 A_0 在圆心位置同时该位置为原点和极点，而剩余的无人机则均匀分布在半径为 $R=100$ 的圆上。在圆上的无人机 A_n 可使用极坐标求出具体如下：

$$\begin{cases} A_n = (r_n, \theta_n) \\ r_n = R & n = 1 \dots 9 \\ \theta_n = \frac{2\pi}{9}(n-1) & n = 1 \dots 9 \end{cases}$$

而无人机 A_n 可使用参数方程把极坐标转为直角坐标，具体如下：

$$\begin{cases} x_n = r_n \cos(\theta_n) \\ y_n = r_n \sin(\theta_n) \end{cases}$$

将第三小问中，给得的10个无人机初始位置极坐标进行转化为坐标表示，得到如下表格：

表3：10架无人机的坐标表

无人机编号	坐标(x,y)	无人机编号	坐标(x,y)
FY00	(0, 0)	FY05	(-92.01, 33.74)
FY01	(100, 0)	FY06	(-105.27, -38.23)
FY02	(74.96, 63.12)	FY07	(-52.39, -90.46)
FY03	(19.04, 110.37)	FY08	(17.30, -96.46)
FY04	(-52.10, 91.16)	FY09	(86.15, -71.57)

根据表格中的数据，以FY00无人机为圆心，半径为100将10架无人机进行可视化得到如下示意图：

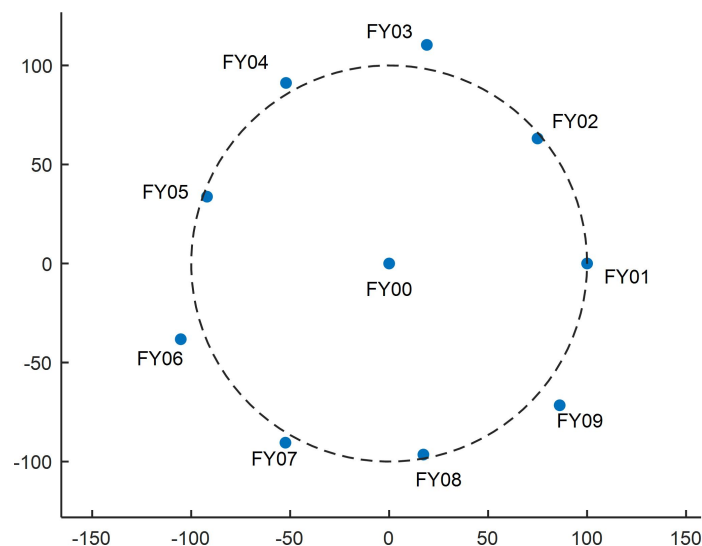


图12：10架无人机初始位置示意图

图中红色的*号为无人机，由该示意图可以非常清晰的看到出FY00和FY01无人机外，其他编号的无人机都出现了不同程度的偏差，其中无人机编号为FY03、FY06、FY09相对于其他无人机偏移量较大。

5.1.3.2 无人机定位计算

当我们选择无人机发射电磁波信号后，依据第一问无人机定位解析式即可求解出待定位无人机的极坐标。但是发射电磁波信号的无人机与理想位置有误差，而待定位无人机计算本身极坐标时，是以发射电磁波信号无人机的理想位置进行计算的。故会产生新的误差（待定位无人机与理想点位的距离），新误差可能变大，也有可能变小。下列计算公式为待测无人机定位解析式：

$$\left\{ \begin{array}{l} \beta_k = \arctan \frac{B_{ijk} (\cos \theta_i - \cos \theta_j) \mp C_{ijk} \sin \theta_i \pm D_{ijk} \sin \theta_j}{B_{ijk} (\sin \theta_j - \sin \theta_i) \mp C_{ijk} \cos \theta_i \pm D_{ijk} \cos \theta_j} \\ r_k = \frac{R}{\sin(\alpha_{0ki})} \sin(\alpha_{0ki} + \alpha_{i0k}) \\ B_{ijk} = \sin \alpha_{0ki} \sin \alpha_{0kj} \\ C_{ijk} = \sin \alpha_{0kj} \cos \alpha_{0ki} \\ D_{ijk} = \sin \alpha_{0ki} \cos \alpha_{0kj} \end{array} \right.$$

其中， β_k 为待定位无人机 A_k 计算的极角， r_k 无人机 A_k 计算的极半径。 R 为圆形编队的半径。而选取的 i, j 编号的无人机为发射电磁波信号无人机， θ_i 为无人机 A_i 理想位置的极角， θ_j 为无人机 A_j 理想位置的极角。而对于角 α 则为待定位无人机 A_k 接收到的角度信号。

在上述解析式中有符号是未确定的如 C_{ijk} 前的负正号， D_{ijk} 前的正负号，符号的确定与待定位无人机 A_k 的位置有关。当我们选取 A_i 、 A_j 无人机后，以 A_0 为起点画一条到无人机 A_i 和 A_j 的直线。

以直线 A_0A_i 对坐标系进行划分两个区域，以 $\overrightarrow{A_0A_i}$ 为轴，在轴的左区域为正区域，在正区域的待定位无人机 A_k ，极角计算公式的 C_{ijk} 前符号取负号，否则取正号。同理以直线 A_0A_j 划分坐标系两个区域，以 $\overrightarrow{A_0A_j}$ 为轴，在轴的左区域为正区域，在正区域的待定位无人机 A_k ，极角计算公式的 D_{ijk} 前符号取正号，否则取负号。如无人机 A_k 在以 $\overrightarrow{A_0A_i}$ 为轴左部正区域空间，具体如下图：

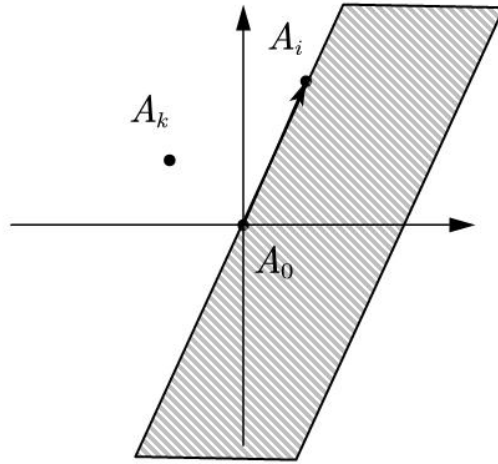


图13: 无人机区域划分

5.1.3.3 调整无人机的位置

当我们计算完待定位无人机的坐标后最后一步是调整无人机的位置。位置调整方法可用向量表示，现已知无人机的极坐标，我们把极坐标转为直角坐标，随

后计算该点位到理想点位的调整向量，具体如下：

$$\begin{cases} p_{ij} = (x_{ij}, y_{ij}) \\ x_{ij} = R \cos(\theta_j) - r_{ij} \cos(\beta_{ij}) \\ y_{ij} = R \sin(\theta_j) - r_{ij} \sin(\beta_{ij}) \end{cases} \quad i = 1 \dots a, j = 1 \dots 9$$

其中， p_{ij} 表示第 j 架无人机第 i 次调整的调整向量， β_{ij} 表示第 i 次调整第 j 架无人机的计算极角， r_{ij} 表示第 i 次调整第 j 架无人机的计算极径。

当我们把无人机位置调整后，在无人机的视角中无人机本身的坐标是在理想点位上的。但现实是发射电磁波信号的无人机位置是有误差的，故在现实视角中调整后的无人机仍有误差。设误差矩阵为 E_{ij} ，调整后现实坐标矩阵 g_{ij} ，具体如下：

$$\begin{cases} E_{ij} = |g_{ij} A_j| \\ g_{ij} = g_{i-1j} + p_{ij} \end{cases} \quad i = 1 \dots a, j = 1 \dots 9$$

其中， E_{ij} 表示第 i 次调整后第 j 架无人机的理想距离， g_{ij} 表示第 i 次调整后第 j 架无人机的现实坐标。而无人机结束调整的标志是误差为 0，具体如下：

$$E_{aj} = 0 \quad j = 1 \dots 9$$

5.1.3.4 确定目标

在本题中，为避免外界干扰，无人机应尽可能保持电磁静默，少向外发射电磁波信号。同理在调整无人机位置时也应遵守电磁静默原则，故我们设 f_{ij} 为第 i 次调整选择第 j 号无人机发射电磁波信号，目标为无人机调整后总发射无人机数最小，可以确立的目标如下：

$$\text{Min } N = \sum_{i=1}^a \sum_{j=0}^9 f_{ij}$$

其中， f_{ij} 为第 i 次调整第 j 号无人机发射信号， a 为调整的总次数。

5.1.3.5 约束条件

为了能够使无人机调整后总发射无人机数最小，需对各个条件进行约束：

(1) 对“0-1”变量的约束

考虑到无人机在调整过程中，需要选择部分无人机发射信号去定位发生偏离的无人机，则需要对第 i 次调整是否选择第 j 号无人机发射电磁波信号 f_{ij} 进行“0-1”变量约束，具体约束如下：

$$f_{ij} \in \{0, 1\} \quad i = 1 \dots a, j = 0 \dots 9$$

(2) 对发射信号无人机架数的约束

由题目中可知，无人机在每次调整的过程中，最多选取4架无人机发射电磁波信号。则可以得知第 j 号无人机发射电磁波信号 f_{ij} 之和大于1架无人机小于4架无人机，具体约束如下：

$$1 \leq \sum_{j=0}^9 f_{ij} \leq 4 \quad i = 1 \dots a$$

在无人机选择的过程中，圆心区域的无人机是必须选择的，也就是编号为FY00的无人机，第 i 次调整是否选择第0号无人机发射电磁波信号 f_{ij} 我们直接进行赋值为1，具体约束如下：

$$f_{ij} = 1 \quad i = 1 \dots a; j = 0$$

5.1.3.5 无人机最优调整方案模型I建立

依据我们对编队理想点的确立，对最优调整目标的设立，对选择发射电磁波信号无人机条件的分析，对无人机定位算法的分析，对无人机调整向量的方向，对无人机误差分析，建立以下模型：

$$\begin{aligned} \text{Min} \quad N &= \sum_{i=1}^a \sum_{j=0}^9 f_{ij} \\ \text{s.t.} \quad &\begin{cases} \beta_{ik} = \arctan \frac{B_{ijk} (\cos \theta_i - \cos \theta_j) \mp C_{ijk} \sin \theta_i \pm D_{ijk} \sin \theta_j}{B_{ijk} (\sin \theta_j - \sin \theta_i) \mp C_{ijk} \cos \theta_i \pm D_{ijk} \cos \theta_j} \\ r_{ik} = \frac{R}{\sin(\alpha_{0ki})} \sin(\alpha_{0ki} + \alpha_{i0k}) \\ B_{ijk} = \sin \alpha_{0ki} \sin \alpha_{0kj} \\ C_{ijk} = \sin \alpha_{0kj} \cos \alpha_{0ki} \\ D_{ijk} = \sin \alpha_{0ki} \cos \alpha_{0kj} \\ 1 \leq \sum_{j=0}^9 f_{ij} \leq 4 \quad i = 1 \dots a \\ f_{i0} = 1 \quad i = 1 \dots a \\ p_{ij} = (x_{ij}, y_{ij}) \\ x_{ij} = R \cos(\theta_j) - r_{ij} \cos(\beta_{ij}) \quad i = 1 \dots a, j = 1 \dots 9 \\ y_{ij} = R \sin(\theta_j) - r_{ij} \sin(\beta_{ij}) \quad i = 1 \dots a, j = 1 \dots 9 \end{cases} \end{aligned}$$

$$s.t. \begin{cases} E_{ij} = \left| \overline{g_{ij} A_j} \right| & i = 1 \dots a, j = 1 \dots 9 \\ g_{ij} = g_{i-1j} + p_{ij} & i = 1 \dots a, j = 1 \dots 9 \\ E_{aj} = 0 & j = 1 \dots 9 \\ f_{ij} \in \{0,1\} & i = 1 \dots a, j = 0 \dots 9 \end{cases}$$

5.1.3.6 基于贪心思想求解模型算法设计的具体描述

根据以上建立的最优化模型，基于贪心算法的思想，设计相对应的求解算法，具体算法步骤如下：

Step1: 我们依据编队要求生成无人机各理想点位直角坐标和极坐标用矩阵 *dronecartesian*、*dronepolar* 存储。创建矩阵 f_{ij} 记录每次调整的发射无人机，元胞数组 g 记录无人机真实的直角坐标和极坐标，元胞数组 *polcompute* 用于记录无人机计算得到的极坐标，创建误差矩阵 E_{ij} 记录每次调整后各无人机距离理想点的误差，创建元胞数组 p 记录每次无人机的调整向量，设置程序最大迭代次数 a 。

Step2: 数据初始化。无人机初始坐标数据由题目可得，我们把无人机初始坐标导入元胞数组 g 中。设置固定发射无人机为 FY00、FY01。以及设置程序最大迭代次数为 10 次。

Step3: 程序用循环语句开始迭代，每次迭代要从编号 FY02~FY09 中 7 架无人机选取 1 架无人机当做发射无人机，选取的标准是基于贪心思想选取的即选取离理想点最近的无人机。选取后开始计算无人机的极坐标，我们依据向量点乘式子得到待定位无人机接收到夹角的信息。随后依据第一题第一问推导出的四个解析式分别计算待定位无人机的 $\tan(\theta)$ 值，依据四个区域限制角度排除其中 3 个错误错误的 $\tan(\theta)$ 值，利用 \arctan 函数反解出无人机的极角（要补全其缺失的角度）。最后依据正弦定理求出其极径。

Step4: 由于带定位无人机计算自身坐标时是假设发射电磁波信号无人机是无偏差的，但现实是发射电磁波信号无人机是有偏差的，故导致待定位无人机计算得出的坐标值是有误差的。我们依据调整向量的计算公式可得出无人机的调整方案，无人机做出相应的移位后的记录其当前的真实坐标，用于下次迭代。

Step5: 最后依据距离公式可得出每次调整后离理想点位的误差。我们对这次调整后计算出的误差做求和，当求和值小于一个预设值（0.0001）时结束迭代。

5.1.3.7 具体结果

综上所述，根据 5.1.3.6 的具体算法求解步骤，利用 *Matlab* 编程实现，求得无人机位置的调整方案，通过计算我们调整的总次数为 5 次，并且我们每次调整选择的无人机架数为 3 次。首先我们给出 5 次调整分别选择了哪些编号的无人机，表格如下所示：

表4：5次调整选择无人机编号

调整次数	1 号无人机	2 号无人机	3 号无人机
1	FY00	FY01	FY02
2	FY00	FY01	FY06
3	FY00	FY01	FY02
4	FY00	FY01	FY03
5	FY00	FY01	FY05

从表格中可以得到，选择第一架和第二架无人机基本都是固定的，本问中给出的坐标可以得到，这2架无人机的位置是无偏差的，所以基于这2架无人机进行定位再选择一架无人机对其进行定位调整，选择的无人机为FY02、FY06、FY03、FY05分别进行五次调整。

通过计算出每一次调整后与准确点的误差，来观察我们调整的情况，我们通过绘制每一次调整的折线图来观察误差变化的情况。折线图如下所示：

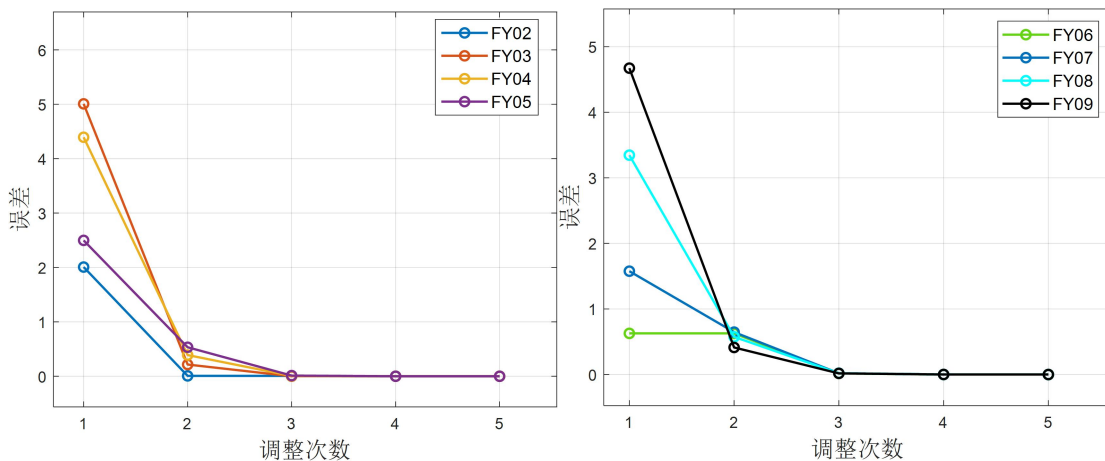


图14：FY02-FY09编号无人机误差折线图

图14中2幅折线图呈现的是编号为FY02-FY09经过5次调整过后距离最终点位的误差，其中FYF01和FY00是位置准确，所以不需要进行调整。上图可以非常清晰的看到，每一次进行调整误差都在减小，说明我们的调整没有错误，无人机都在向正确的位置进行调整，当调整次数到达3次时，误差接近与0，由于时绘制折线图的原因，小于0的部分不能很好的呈现，其实我们调整的结果到第五次，误差还是没有达到0。

最后，给出经过每次调整后的各个无人机的坐标，调整坐标如下：

表5：各编号无人机4次调整坐标表

	第一次调整		第二次调整	
无人机编号	x	y	x	y

<i>FY02</i>	74.96229729	63.12411572	74.96229729	63.12411572
<i>FY03</i>	12.80440677	96.41170529	12.80440677	96.41170529
<i>FY04</i>	-50.18541839	82.21250716	-50.18541839	82.21250716
<i>FY05</i>	-91.81389755	32.93949146	-91.81389755	32.93949146
<i>FY06</i>	-93.42896811	-33.8828741	-93.42896811	-33.8828741
<i>FY07</i>	-49.98701687	-88.17883974	-49.98701687	-88.17883974
<i>FY08</i>	20.20471735	-100.2499747	20.20471735	-100.2499747
<i>FY09</i>	80.3374451	-61.46792367	80.3374451	-61.46792367
	第三次调整		第四次调整	
<i>FY02</i>	76.61046835	64.27552266	76.6050884	64.278389
<i>FY03</i>	17.36529404	98.48105232	17.36529404	98.48105232
<i>FY04</i>	-49.99997682	86.59622203	-50.00000002	86.60283606
<i>FY05</i>	-93.95852571	34.19584105	-93.9692644	34.20201567
<i>FY06</i>	-93.95456131	-34.19357091	-93.96901042	-34.20186906
<i>FY07</i>	-49.99987241	-86.58301852	-49.9999999	-86.60199159
<i>FY08</i>	17.34776839	-98.47073058	17.36417626	-98.48040467
<i>FY09</i>	76.58908818	-64.28747862	76.60371354	-64.27918256

根据上表我们可以得出无人机每次调整的位置，结合误差分析图，在第三次调整的时候误差已经接近与0，说明编号FY02-FY09已经接近与正确位置，也就是距离圆心编号为FY00的无人机距离为100m，当然有了上表，我们也可以计算出各编号无人机到圆心的距离，根据是否为100m判断是否调整正确。综上所述，我们采用3次调整作为我们的调整方案。下图是对调整结果进行可视化：

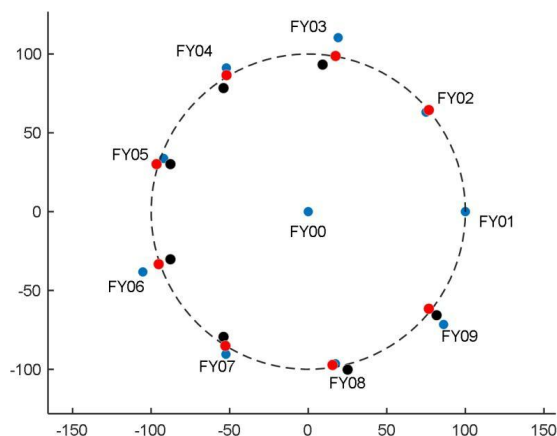


图15：无人机调整示意图

我们绘制经过第二次第三次调整的示意图，上图中蓝色点为初始偏差点位，黑色点位为第二次调整点位，红色点位为最终调整完成时的点位。

5.2 问题2：当编队为锥形时的调度方案

对于第二题，由于在无人机实际飞行过程中，无人机的编队队形可以不局限于同一个平面，而是在三维空间中程规律分布，同时需要考虑纯方位无源定位的情形，设计无人机位置调整方案，而对于本问，需要给出无人机锥形编队队形下的无人机位置的确定以及对应调整方案。

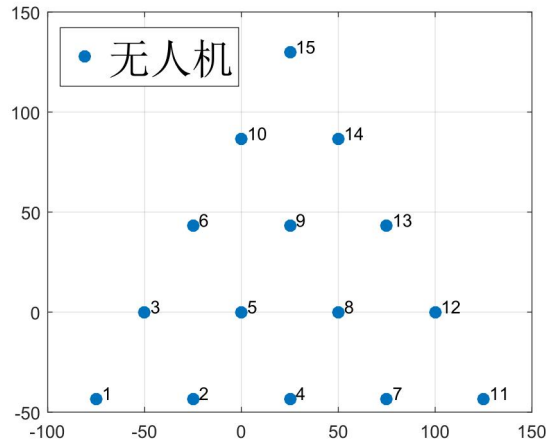


图16：无人机编号及其位置示意图

5.2.1 锥形编队位置预设

在本题中，将无人机的编队设置为锥形编队，需要在编队飞行过程纯方位无源定位的方法调整无人机的位置，以及当遇到障碍物对于无人机的高度位置进行调整时的方案，其中一架无人机在锥形位置 A_1 时为无人机的领路人，考虑到无线波在传播过程中会有消耗，需要让无人机的位置尽可能的靠近坐标轴，且选取的原点的位置在无人机上，尽量使该无人机进行发射信号调整无人机的位置，因此，我们寻找该锥形编队的内心，寻找得到离该锥形内心最近的无人机作为坐标原点，该锥形编队的内心具体公式如下：

$$(x, y) = \left(\frac{ax_1 + bx_2 + cx_3}{a + b + c}, \frac{ay_1 + by_2 + cy_3}{a + b + c} \right)$$

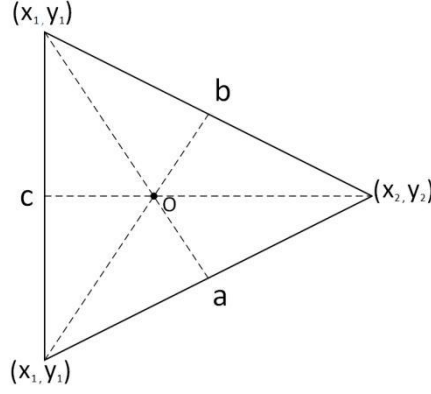


图 17：内心示意图

根据上述公式计算得到距离锥形内心最近的点为 A_5, A_8, A_9 ，因此我们设置其中一架无人机 A_5 为坐标轴原点，而剩余的无人机与相邻的无人机都间距 $50m$ ，在锥形上的无人机 A_n 可使用极坐标求出具体如下：

$$A_n = \left(\sqrt{(x_n - x_5)^2 + (y_n - y_5)^2}, \theta_n, \varphi_n \right) \quad n = 1 \dots 15$$

而无人机 A_n 可使用参数方程把极坐标转为直角坐标，具体如下：

$$\begin{cases} x_n = r_n \sin(\theta) \cos(\varphi) \\ y_n = r_n \sin(\theta) \sin(\varphi) \\ z_n = r_n \sin(\theta) \end{cases}$$

5.2.2 无人机位置调整

对于无人机进行水平方向的定位后，最后一步是调整无人机的水平位置，因为无人机基于自身感知的高度信息，均保持在同一个高度上飞行。位置调整方法可用向量表示，现在已知无人机的极坐标，我们把极坐标转为直角坐标，随后计算该点位到理想点位的调整向量，具体如下：

$$\begin{cases} p_{ij} = (x_{ij}, y_{ij}) \\ x_{ij} = R \cos(\theta_j) - r_{ij} \cos(\beta_{ij}) \\ y_{ij} = R \sin(\theta_j) - r_{ij} \sin(\beta_{ij}) \end{cases} \quad i = 1 \dots a, j = 1 \dots 9$$

其中， p_{ij} 表示第 j 架无人机第 i 次调整的调整向量， β_{ij} 表示第 i 次调整第 j 架无人机的计算极角， r_{ij} 表示第 i 次调整第 j 架无人机的计算极径。

当我们把无人机位置调整后，在无人机的视角中无人机本身的坐标是在理想点位上的。但现实是发射电磁波信号的无人机位置是有误差的，故在现实视角中调整后的无人机仍有误差。设误差矩阵为 E_{ij} ，调整后现实坐标矩阵 g_{ij} ，具体如下：

$$\begin{cases} E_{ij} = |g_{ij} - A_j| & i = 1 \dots a, j = 1 \dots 9 \\ g_{ij} = g_{i-1j} + p_{ij} & i = 1 \dots a, j = 1 \dots 9 \end{cases}$$

其中, E_{ij} 表示第 i 次调整后第 j 架无人机与理想点位的距离, g_{ij} 表示第 i 次调整后第 j 架无人机的现实坐标。而无人机结束调整的标志是误差为 0, 具体如下:

$$E_{aj} = 0 \quad j = 1 \dots 9$$

5.2.3 遇到障碍物进行高度调整

在本题中, 我们需要考虑在遇到障碍物时, 为了规避障碍物对无人机高度的调整, 为了躲避与障碍物的碰撞, 我们应求得领头的无人机应到达的高度, 为了保持编队队形整齐, 因此编队在水平位置上仍然需要保持位置。

现已知需要避让的障碍物与无人机编队相对高度相差, 我们通过使用旋转矩阵对于无人机机群的高度进行调整, 我们可以得到无人机编队经过高度调整后的坐标, 具体公式如下:

$$\begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} \quad n = 1 \dots 15$$

5.2.4 修改模型 I 的目标

本问的目标, 沿用第一问的目标, 对目标的下标进行修改, 因为无人机的架数不同。同理在调整无人机位置时也应遵守电磁静默原则, 故我们设 f_{ij} 为第 i 次调整选择第 j 号无人机发射电磁波信号, 目标为无人机调整后总发射无人机数最小, 可确定的目标如下:

$$\text{Min} \quad N = \sum_{i=1}^a \sum_{j=0}^{15} f_{ij}$$

其中, f_{ij} 为第 i 次调整第 j 号无人机发射信号, a 为调整的总次数。

5.2.5 修改模型 I 的约束条件

为了能够使无人机调整后总发射无人机数最小, 需对各个条件进行约束:

(1) 对“0-1”变量的约束

考虑到无人机在调整过程中, 需要选择部分无人机发射信号去定位发生偏离的无人机, 则需要对第 i 次调整是否选择第 j 号无人机发射电磁波信号 f_{ij} 进行“0-1”变量约束, 具体约束如下:

$$f_{ij} \in \{0, 1\} \quad i = 1 \dots a, j = 0 \dots 15$$

(2) 对发射信号无人机架数的约束

由题目中可知，无人机在每次调整的过程中，最多选取3架无人机发射电磁波信号。则可以得知第 j 号无人机发射电磁波信号 f_{ij} 之和小于3架无人机，具体约束如下：

$$\sum_{j=0}^{15} f_{ij} \leq 3 \quad i=1 \dots a$$

5.2.5 模型 II 的建立与求解

依据我们对编队理想点的确立，对最优调整目标的设立，对选择发射电磁波信号无人机条件的分析，对无人机定位算法的分析，对无人机调整向量的方向，对无人机进行高度的调整，对无人机误差分析，建立以下模型：

$$\begin{aligned} \text{Min} \quad N &= \sum_{i=1}^a \sum_{j=0}^{15} f_{ij} \\ \text{s.t.} \quad &\left\{ \begin{aligned} \beta_{ik} &= \arctan \frac{B_{ijk} (\cos \theta_i - \cos \theta_j) \mp C_{ijk} \sin \theta_i \pm D_{ijk} \sin \theta_j}{B_{ijk} (\sin \theta_j - \sin \theta_i) \mp C_{ijk} \cos \theta_i \pm D_{ijk} \cos \theta_j} \\ r_{ik} &= \frac{R}{\sin(\alpha_{0ki})} \sin(\alpha_{0ki} + \alpha_{i0k}) \\ B_{ijk} &= \sin \alpha_{0ki} \sin \alpha_{0kj} \\ C_{ijk} &= \sin \alpha_{0kj} \cos \alpha_{0ki} \\ D_{ijk} &= \sin \alpha_{0ki} \cos \alpha_{0kj} \\ 3 &\leq \sum_{j=0}^9 f_{ij} \quad i=1 \dots a \\ p_{ij} &= (x_{ij}, y_{ij}, z_{ij}) \\ x_{ij} &= R_j \cos(\theta_j) - r_{ij} \cos(\beta_{ij}) \quad i=1 \dots a, j=1 \dots 15 \\ y_{ij} &= R_j \sin(\theta_j) - r_{ij} \sin(\beta_{ij}) \quad i=1 \dots a, j=1 \dots 15 \\ E_{ij} &= |\overline{g_{ij} A_j}| \quad i=1 \dots a, j=1 \dots 15 \\ g_{ij} &= g_{i-1j} + p_{ij} \quad i=1 \dots a, j=1 \dots 15 \\ E_{aj} &= 0 \quad j=1 \dots 15 \\ f_{ij} &\in \{0, 1\} \quad i=1 \dots a, j=0 \dots 15 \end{aligned} \right. \end{aligned}$$

5.2.6 基于贪心思想求解模型算法设计的具体描述

根据以上建立的最优化模型，基于贪心算法的思想，设计相对应的求解算法，

具体算法步骤如下：

Step1: 我们依据编队要求生成无人机各理想点位直角坐标和极坐标用矩阵 *dronecartesian*、*dronepolar* 存储。创建矩阵 f_{ij} 记录每次调整的发射无人机，元胞数组 *g* 记录无人机真实的直角坐标和极坐标，元胞数组 *polcompute* 用于记录无人机计算得到的极坐标，创建误差矩阵 E_{ij} 记录每次调整后各无人机距离理想点的误差，创建元胞数组 *p* 记录每次无人机的调整向量，设置程序最大迭代次数 *a*。

Step2: 数据初始化。已知锥形编队无人机初始理想坐标数据，再由随机函数得知，我们把无人机初始坐标导入元胞数组 *g* 中，设置程序最大迭代次数为 10 次。

Step3: 程序用循环语句开始迭代，每次迭代要从编号 FY01~FY15 中 15 架无人机选取 3 架无人机当做发射无人机，选取的标准是基于贪心思想选取的即选取离理想点最近的无人机。选取后开始计算无人机的极坐标，我们依据向量点乘式子得到待定位无人机接收到夹角的信息。随后依据第一题第一问推导出的四个解析式分别计算待定位无人机的 $\tan(\theta)$ 值，依据四个区域限制角度排除其中 3 个错误的 $\tan(\theta)$ 值，利用 \arctan 函数反解出无人机的极角（要补全其缺失的角度）。最后依据正弦定理求出其极径。

Step4: 由于带定位无人机计算自身坐标时是假设发射电磁波信号无人机是无偏差的，但现实是发射电磁波信号无人机是有偏差的，故导致待定位无人机计算得出的坐标值是有误差的。我们依据调整向量的计算公式可得出无人机的调整方案，无人机做出相应的移位后的记录其当前的真实坐标，用于下次迭代。

Step5: 最后依据距离公式可得出每次调整后离理想点位的误差。我们对这次调整后计算出的误差做求和，当求和值小于一个预设值（0.0001）时结束迭代。

5.2.7 具体结果

综上所述，根据 5.1.3.6 的具体算法求解步骤，利用 *Matlab* 编程实现，求得锥形无人机编队发生偏差时的位置调整方案，求解得到最小调整次数为 2 次。第一次选择发射信号的无人机编号为 FY01、FY05、FY06，第二次选择发射信号无人机的编号为 FY01、FY05、FY08。具体调度方案如下表所示：

表 6：调度方案部分表

无人机	理想坐标	实际坐标	第一次调整	第二次调整
1	(-75 -43.3)	(-75 -43.3)	(-75 -43.3)	(-75 -43.3)
2	(-25 -43.3)	(-19.9 -38.2)	(-24.8 -43.1)	(-75 -43.2)
3	(-50 -43.3)	(-45.6 -42.4)	(-49.6 -42.9)	(-49.9 -43.3)
4	(25 -43.3)	(27.5 -40.8)	(25.5 -43.9)	(25.1 -43.3)

...
13	(75 -43.3)	(75.6 -43.9)	(75.6 -43.9)	(75 -43.4)
14	(50 86.6)	(51.6 88.2)	(50.6 87.2)	(50 86.7)
15	(25 129.9)	(28.3 133.2)	(25.5 130.4)	(25.2 130)

六、模型评价与推广

6.1 模型优点

- (1) 采用最优化模型求得最优解。
- (2) 通过几何关系进行推导定位模型，精准度高。
- (3) 基贪心思想求解最优调整方案

6.2 模型缺点

- (1) 过于理想化的假设，在实际情况中复杂的环境因素会对无人机编队造成较大影响。

6.3 模型推广

无人机在灾害救援、生活、农业、军事方面的应用越来越广泛，无人机的纯方位无源定位可以避免外界的干扰，提高无人机编队在飞行过程中的稳定性，且由飞行中的无人机发射信号进行定位，可以实时进行编队的调整，更快的实现队形的变换在军事作战、灾害救援中更加适应多变的环境，本文所研究的定位模型通过几何关系进行推导，精准度高能更好的做出定位。

七、参考文献

- [1] 李锐博恩 无源定位学习总结（一）
https://blog.csdn.net/Reborn_Lee/article/details/82468058
- [2] 旋转矩阵的推导
<https://baike.baidu.com/item/%E7%9F%A9%E9%98%B5/18069>

附录

一、支撑材料文件列表

1. 第一题第一小问模型 I *matlab* 代码

```
%%  
%三点 ABC 计算角度  
%输入 ABC 三点直角坐标  
%输出角 BAC 夹角  
function [aerfa]= aerfaCompute (A,B,C)  
%向量  
AB=[];  
AC=[];  
AB(1)=B(1)-A(1);  
AB(2)=B(2)-A(2);  
AC(1)=C(1)-A(1);  
AC(2)=C(2)-A(2);  
ABAC=AB(1)*AC(1)+AB(2)*AC(2);  
LAB=(AB(1)^2+AB(2)^2)^0.5;  
LAC=(AC(1)^2+AC(2)^2)^0.5;  
aerfa=acos(ABAC/(LAB*LAC));  
end  
  
%%  
%计算两点的距离  
%输入 AB 两点点直角坐标  
%输出 LAB  
function [L]= linkCompute (A,B)  
AB(1)=A(1)-B(1);  
AB(2)=A(2)-B(2);  
L=(AB(1)^2+AB(2)^2)^0.5;  
end  
  
%%  
%生成测试用例
```

```

function [test]=testCreat()
test=[];%测试用例第 1, 2 列为直角坐标(x, y), 第 3, 4 列为极坐标(r,  $\theta$ )
%圆形编队无人机坐标与极坐标
dronepolar=zeros(10,2);%无人机极地坐标
dronecartesian=zeros(10,2);%无人机直角坐标
r=100;%圆形半径
xita=2*pi/9;%角度间隔
for i=2:10;
    dronepolar(i,1)=r;
    dronepolar(i,2)=xita*(i-2);
    dronecartesian(i,1)=dronepolar(i,1)*cos(dronepolar(i,2));
    dronecartesian(i,2)=dronepolar(i,1)*sin(dronepolar(i,2));
end
%在标准点附近生成随机点
for i=1:5000
    n=unidrnd(9)+1;
    test(i,1)=dronecartesian(n,1)+rand()*20-10;%x 坐标
    test(i,2)=dronecartesian(n,2)+rand()*20-10;%y 坐标
    [test(i,4), test(i,3)]=cart2pol(test(i,1), test(i,2));%极坐标
    test(i,5)=rad2deg(test(i,4));
    test(i,5)=mod(test(i,5), 360);%角度
    test(i,6)=tan(test(i,4));%tan 值
end
end

%%
%计算两圆交点
%输入圆心直角坐标 O1, O2, 半径 r1, 半径 r2, 剔除点 p1
%输出交点
function [p2]= intCompute (O1,O2,r1,r2,p1)
syms x y
a1=O1(1);a2=O2(1);b1=O1(2);b2=O2(2);%给出圆参数
[x,y]=solve((x-a1)^2+(y-b1)^2-r1^2, (x-a2)^2+(y-b2)^2-r2^2);
if x(1)~=p1(1)&&y(1)~=p1(2)

```

```

        p2(1)=double(x(1));
        p2(2)=double(y(1));
elseif x(2)~=p1(1)&&y(2)~=p1(2)
        p2(1)=double(x(2));
        p2(2)=double(y(2));
else
        p2(1)=p1(1);
        p2(2)=p1(1);
end
end

%%
%圆形编队无人机坐标与极坐标
dronepolar=zeros(10,2);%无人机极地坐标
dronecartesian=zeros(10,2);%无人机直角坐标
r=100;%圆形半径
xita=2*pi/9;%角度间隔
n=1;
for i=2:10;
    dronepolar(i,1)=r;
    dronepolar(i,2)=xita*(i-2);
    dronecartesian(i,1)=dronepolar(i,1)*cos(dronepolar(i,2));
    dronecartesian(i,2)=dronepolar(i,1)*sin(dronepolar(i,2));
end
%%
%验证推导正确性
test=testCreat();
A0=1;%无人机 A0
A1=2;%无人机 A1
A2=3;%无人机 A2
A3=zeros(size(test,1),4);
L01=linkCompute(dronecartesian(A0,:),dronecartesian(A1,:));
L12=linkCompute(dronecartesian(A1,:),dronecartesian(A2,:));

```



```

L02=linkCompute(dronecartesian(A0,:), dronecartesian(A2,:));
xita3=[];
for i=1:size(test,1)
    %计算 k1,k2,k3

    arf031=aerfaCompute(test(i,1:2), dronecartesian(A0,:), dronecartesian(A
    1,:));

    arf132=aerfaCompute(test(i,1:2), dronecartesian(A1,:), dronecartesian(A
    2,:));

    arf032=aerfaCompute(test(i,1:2), dronecartesian(A0,:), dronecartesian(A
    2,:));

    k1=L01/sin(arf031);
    k2=L12/sin(arf132);
    k3=L02/sin(arf032);
    %计算极角
    for j=1:4
        if j==1 || j==2
            a=-1;
        else
            a=1;
        end
        if j==1 || j==3
            b=1;
        else
            b=-1;
        end

        xita3(i,j)=(sin(arf031)*sin(arf032)*(cos(dronepolar(A1,2))-cos(dronep
        olar(A2,2)))+a*sin(arf032)*cos(arf031)*sin(dronepolar(A1,2))+b*sin(ar
        f031)*cos(arf032)*sin(dronepolar(A2,2)))/(sin(arf031)*sin(arf032)*(si
        n(dronepolar(A2,2))-sin(dronepolar(A1,2)))+a*sin(arf032)*cos(arf031)*
        cos(dronepolar(A1,2))+b*sin(arf031)*cos(arf032)*cos(dronepolar(A2,2)))
    end
end

```

```

;
end
%排除错误极角
min=100;
for j=1:4
    if abs(xita3(i,j)-test(i,6))<min
        A3(i,2)=xita3(i,j);%tan(xita3)
        min=abs(xita3(i,j)-test(i,6));%计算误差
        A3(i,3)=min;%记录误差
        A3(i,4)=j;%记录角度区域
    end
end
%转极角
if A3(i,4)==1||A3(i,4)==2
    A3(i,5)=atan(A3(i,2));%极角
    if A3(i,5)<0
        A3(i,5)=A3(i,5)+pi;
    end
    xita3(i,5)=mod(rad2deg(A3(i,5)),360);
    arf103=A3(i,5);
else
    A3(i,5)=atan(A3(i,2));
    if A3(i,5)>0
        A3(i,5)=A3(i,5)+pi;
    end
    xita3(i,5)=mod(rad2deg(A3(i,5)),360);
    arf103=-A3(i,5);
end
%计算极径
A3(i,1)=k1*sin(arf031+arf103);
end
for i=1:size(test,1)
    A3(i,6)=A3(i,1)*cos(A3(i,5));%x 坐标
    A3(i,7)=A3(i,1)*sin(A3(i,5));%y 坐标
end

```

```

end
hold on
%plot(test(:,1),test(:,2),'.')
plot(A3(:,6),A3(:,7),'.')
%plot(dronecartesian(1:3,1),dronecartesian(1:3,2))

```

2. 模型 III matlab 代码

```

clear;
s1=xlsread('C:\Users\1\Desktop\圆形编队数据.xlsx','B2:C11');%坐标

%对角度遍历
R=100;
e=zeros(100,1);%标准位置时与第 0, k 架的夹角
e1=zeros(100,1);%标准位置时与第 1, k 架的夹角
f=zeros(100,1000);%误差位置中与第 0, k 架的夹角
g=zeros(10,2);%统计最大角度值与最小角度值
m=zeros(10000,2000);
o=1,p=1;
for i=3:10
    e(i)=aerfaCompute([s1(9,1:2)],[0,0],[s1(i,1:2)]);%计算标准位置时
    与第 0, k 架的夹角
    e1(i)=aerfaCompute([s1(9,1:2)],[s1(2,1:2)],[s1(i,1:2)]);%计算标准
    位置时与第 0, k 架的夹角
end

for i=3:10
    k=1;
    for j=-15:0.1:15
        for h=-15:0.1:15
            s2(1,1)=s1(9,1)+h;s2(1,2)=s1(9,2)+j;%对接收信号的无人机位
            置改变
            count=aerfaCompute([s2(1,1:2)],[0,0],[s1(2,1:2)]);

            m(o,p)=(s2(1,1)-R/2)^2+(s2(1,2)+R/2/cos(count))^2-(R/2/sin(count))^2;
            p=p+1;
            if (s2(1,1)-R/2)^2+(s2(1,2)+R/2/cos(count))^2-
                (R/2/sin(count))^2<0.01;%圆的方程
                f(i,k)=aerfaCompute([s2(1,1:2)],[0,0],[s1(i,1:2)]);%
                计算误差位置时与第 0, k 架的夹角

                f1(i,k)=aerfaCompute([s2(1,1:2)],[s1(2,1:2)],[s1(i,1:
                2)]);%计算误差位置时与第 1, k 架的夹角
            end
        end
    end
end

```

```

            k=k+1;
        end
    end
    o=o+1;p=1;
end
end
for i=3:10
    g(i,1)=min(f(i,:));%找到夹角 1 中最小的值
    g(i,2)=max(f(i,:));%找到夹角 1 中最大的值
    g1(i,1)=min(f1(i,:));%找到夹角 2 中最小的值
    g1(i,2)=max(f1(i,:));%找到夹角 2 中最大的值
end
%%
%三点 ABC 计算角度
%输入 ABC 三点直角坐标
%输出角 BAC 夹角
function [aerfa]= aerfaCompute (A,B,C)
%向量
AB=[];
AC=[];
AB(1)=B(1)-A(1);
AB(2)=B(2)-A(2);
AC(1)=C(1)-A(1);
AC(2)=C(2)-A(2);
ABAC=AB(1)*AC(1)+AB(2)*AC(2);
LAB=(AB(1)^2+AB(2)^2)^0.5;
LAC=(AC(1)^2+AC(2)^2)^0.5;
aerfa=acos(ABAC/(LAB*LAC));
end

```

3. 第一题第三小问模型 III matlab 代码

```

%无人机位置调整方案
%%
%圆形编队无人机坐标与极坐标
dronepolar=zeros(10,2);%无人机极地坐标
dronecartesian=zeros(10,2);%无人机直角坐标
r=100;%圆形半径
xita=2*pi/9;%角度间隔
n=1;
for i=2:10;
    dronepolar(i,1)=r;

```

```

    dronepolar(i,2)=xita*(i-2);
    dronecartesian(i,1)=dronepolar(i,1)*cos(dronepolar(i,2));
    dronecartesian(i,2)=dronepolar(i,1)*sin(dronepolar(i,2));
end
%%
%无人机位置调整
maxa=10;%设置最大迭代次数
f=[];%记录每次迭代选取的点位
g=cell(maxa+1,2);%无人机实际直角坐标和极坐标
polcompute=cell(maxa,1);%无人机计算极坐标
error=zeros(maxa,size(dronecartesian,1));%误差矩阵
p=cell(maxa,1);%调整向量
%数据初始化
g{1,1}=xlsread('D:\桌面\圆形编队数据.xlsx','Sheet2','B2:C11');
g{1,2}=xlsread('D:\桌面\圆形编队数据.xlsx','Sheet2','D2:E11');
A0=1;%固定选择点
A1=2;%固定选择点
A2=-1;%设置遍历点
%%
for i=1:maxa
    %%
    %贪心思想选取点位（离理想点最近）
    f(i,1)=A0;%设置固定点
    f(i,2)=A1;%设置固定点
    min=100;%无人机误差距离初始化
    for j=3:size(dronecartesian,1)%从 3 开始遍历全部理想点
        L=linkCompute(dronecartesian(j,:),g{i}(j,:));%计算两点距离
        if L<min
            A2=j;
            f(i,3)=A2;
            min=L;
        end
    end
end
%%

```

```

%计算无人机极坐标
for j=1:size(dronecartesian,1)
    dcd=[];
    %判断是否为发射无人机
    if j==A0 || j==A1 || j==A2

[polcompute{i}(j,2),polcompute{i}(j,1)]=cart2pol(g{i,1}(j,1),g{i,1}(j,
2));

        continue;
    end
    %%
    %计算无人机理想距离
    L01=linkCompute(dronecartesian(A0,:),dronecartesian(A1,:));
    L12=linkCompute(dronecartesian(A1,:),dronecartesian(A2,:));
    L02=linkCompute(dronecartesian(A0,:),dronecartesian(A2,:));
    %计算无人机接收到的夹角
    arf031=aerfaCompute(g{i,1}(j,:),g{i,1}(A0,:),g{i,1}(A1,:));
    arf132=aerfaCompute(g{i,1}(j,:),g{i,1}(A1,:),g{i,1}(A2,:));
    arf032=aerfaCompute(g{i,1}(j,:),g{i,1}(A0,:),g{i,1}(A2,:));
    %%
    %计算极角
    xita3=[];%记录极角
    for k=1:4
        if k==1 || k==2
            a=-1;
        else
            a=1;
        end
        if k==1 || k==3
            b=1;
        else
            b=-1;
        end
    end

```

```

xita3(k)=(sin(arf031)*sin(arf032)*(cos(dronepolar(A1,2))-cos(dronepolar(A2,2)))+a*sin(arf032)*cos(arf031)*sin(dronepolar(A1,2))+b*sin(arf031)*cos(arf032)*sin(dronepolar(A2,2)))/(sin(arf031)*sin(arf032)*(sin(dronepolar(A2,2))-sin(dronepolar(A1,2)))+a*sin(arf032)*cos(arf031)*cos(dronepolar(A1,2))+b*sin(arf031)*cos(arf032)*cos(dronepolar(A2,2)));
end
%排除错误极角
min=100;%误差
for k=1:4
    if abs(xita3(k)-tan(g{i,2}(j,2)))<min
        polcompute{i}(j,2)=xita3(k);%tan(xita3)
        xita3(5)=k;%记录角度区域
        min=abs(xita3(k)-tan(g{i,2}(j,2))));%误差更新
    end
end
%转极角
if xita3(5)==1||xita3(5)==2
    polcompute{i}(j,2)=atan(xita3(xita3(5))));%转极角
    if polcompute{i}(j,2)<0
        polcompute{i}(j,2)=polcompute{i}(j,2)+pi;
    end
    arf103=polcompute{i}(j,2);
else
    polcompute{i}(j,2)=atan(xita3(xita3(5))));
    if polcompute{i}(j,2)>0
        polcompute{i}(j,2)=polcompute{i}(j,2)+pi;
    end
    arf103=-polcompute{i}(j,2);
end
%%
%计算极径
k1=L01/sin(arf031);
polcompute{i}(j,1)=k1*sin(arf031+arf103);
end

```

```

%%
%计算调整向量
for j=1:size(dronecartesian,1)

p{i}(j,1)=r*cos(dronepolar(j,2))-polcompute{i}(j,1)*cos(polcompute{i}
(j,2));

p{i}(j,2)=r*sin(dronepolar(j,2))-polcompute{i}(j,1)*sin(polcompute{i}
(j,2));
end
%调整无人机
for j=1:size(dronecartesian,1)
    if j==A0||j==A1||j==A2
        g{i+1,1}(j,1)=g{i,1}(j,1);%x 坐标
        g{i+1,1}(j,2)=g{i,1}(j,2);%y 坐标
        g{i+1,2}(j,1)=g{i,2}(j,1);%r 坐标
        g{i+1,2}(j,2)=g{i,2}(j,2);%θ 坐标
        continue;
    end
    g{i+1,1}(j,1)=g{i,1}(j,1)+p{i}(j,1);%x 坐标
    g{i+1,1}(j,2)=g{i,1}(j,2)+p{i}(j,2);%y 坐标

[g{i+1,2}(j,2),g{i+1,2}(j,1)]=cart2pol(g{i+1,1}(j,1),g{i+1,1}(j,2));
end
%%
%计算误差
for j=1:size(dronecartesian,1)
    error(i,j)=linkCompute(g{i+1,1}(j,:),dronecartesian(j,:));
end
if sum(error(i,:))<0.0001
    break;
end
end
%%

```



```

hold on
for i=1:6
    plot(g{i,1}(:,1),g{i,1}(:,2),'.','markersize',20)
    pause(1)
end

```

3. 第二题 *Matlab* 代码

```

%%
%读取数据
dronepolar=[];%无人机理想点位直角坐标
dronecartesian=[];%无人机理想点位极坐标
dronecartesian=xlsread('D:\桌面\锥形编队数据.xlsx','Sheet1','B2:C16');
dronepolar=xlsread('D:\桌面\锥形编队数据.xlsx','Sheet1','D2:E16');
%plot(dronecartesian(:,1),dronecartesian(:,2),'.')
maxa=2000;%设置最大迭代次数
f=[];%记录每次迭代选取的点位
g=cell(maxa+1,2);%无人机实际直角坐标和极坐标
polcompute=cell(maxa,1);%无人机计算极坐标
error=zeros(maxa,size(dronecartesian,1));%误差矩阵
p=cell(maxa,1);%调整向量
g{1,1}=xlsread('D:\桌面\锥形编队数据.xlsx','Sheet2','B2:C16');
g{1,2}=xlsread('D:\桌面\锥形编队数据.xlsx','Sheet2','D2:E16');
A0=5;%固定选择点
A1=8;%固定选择点
A2=-1;%设置遍历点
%%
for i=1:maxa
    %%
    %贪心思想选取点位（离理想点最近）
    f(i,1)=A0;%设置固定点
    f(i,2)=A1;%设置固定点
    min=100;%无人机误差距离初始化
    for j=1:size(dronecartesian,1)%开始遍历全部理想点

```

```

        if j==A0 || j==A1
            continue;
        end
        L=linkCompute(dronecartesian(j,:),g{i}(j,:));%计算两点距离
        if L<min
            A2=j;
            f(i,3)=A2;
            min=L;
        end
    end
end
%%
%计算无人机极坐标
for j=1:size(dronecartesian,1)
    dcd=[];
    %判断是否为发射无人机
    if j==A0 || j==A1 || j==A2

[polcompute{i}(j,2),polcompute{i}(j,1)]=cart2pol(g{i,1}(j,1),g{i,1}(j,
2));

        continue;
    end
    %%
    %计算无人机理想距离
    L01=linkCompute(dronecartesian(A0,:),dronecartesian(A1,:));
    L12=linkCompute(dronecartesian(A1,:),dronecartesian(A2,:));
    L02=linkCompute(dronecartesian(A0,:),dronecartesian(A2,:));
    %计算无人机接收到的夹角
    arf031=aerfaCompute(g{i,1}(j,:),g{i,1}(A0,:),g{i,1}(A1,:));
    arf132=aerfaCompute(g{i,1}(j,:),g{i,1}(A1,:),g{i,1}(A2,:));
    arf032=aerfaCompute(g{i,1}(j,:),g{i,1}(A0,:),g{i,1}(A2,:));
    %%
    %计算极角
    xita3=[];%记录极角
    for k=1:4

```

```

        if k==1 || k==2
            a=-1;
        else
            a=1;
        end
        if k==1 || k==3
            b=1;
        else
            b=-1;
        end

xita3(k)=(sin(arf031)*sin(arf032)*(cos(dronepolar(A1,2))-cos(dronepolar(A2,2)))+a*sin(arf032)*cos(arf031)*sin(dronepolar(A1,2))+b*sin(arf031)*cos(arf032)*sin(dronepolar(A2,2)))/(sin(arf031)*sin(arf032)*(sin(dronepolar(A2,2))-sin(dronepolar(A1,2)))+a*sin(arf032)*cos(arf031)*cos(dronepolar(A1,2))+b*sin(arf031)*cos(arf032)*cos(dronepolar(A2,2)));
    end
    %排除错误极角
    min=10000;%误差
    for k=1:4
        if abs(xita3(k)-tan(g{i,2}(j,2)))<min
            polcompute{i}(j,2)=xita3(k);%tan(xita3)
            xita3(5)=k;%记录角度区域
            min=abs(xita3(k)-tan(g{i,2}(j,2)));%误差更新
        end
    end
    %转极角
    if xita3(5)==1 || xita3(5)==2
        polcompute{i}(j,2)=atan(xita3(xita3(5)));%转极角
        if polcompute{i}(j,2)<0
            polcompute{i}(j,2)=polcompute{i}(j,2)+pi;
        end
        arf103=polcompute{i}(j,2);
    else

```

```

        polcompute{i}(j,2)=atan(xita3(xita3(5)));
        if polcompute{i}(j,2)>0
            polcompute{i}(j,2)=polcompute{i}(j,2)+pi;
        end
        arf103=-polcompute{i}(j,2);
    end
    %%
    %计算极径
    k1=L01/sin(arf031);
    polcompute{i}(j,1)=k1*sin(arf031+arf103);
end
%%
%计算调整向量
for j=1:size(dronecartesian,1)

p{i}(j,1)=dronepolar(j,1)*cos(dronepolar(j,2))-polcompute{i}(j,1)*cos
(polcompute{i}(j,2));

p{i}(j,2)=dronepolar(j,1)*sin(dronepolar(j,2))-polcompute{i}(j,1)*sin
(polcompute{i}(j,2));

    end
    %调整无人机
    for j=1:size(dronecartesian,1)
        if j==A0||j==A1||j==A2
            g{i+1,1}(j,1)=g{i,1}(j,1);%x 坐标
            g{i+1,1}(j,2)=g{i,1}(j,2);%y 坐标
            g{i+1,2}(j,1)=g{i,2}(j,1);%r 坐标
            g{i+1,2}(j,2)=g{i,2}(j,2);%θ 坐标
            continue;
        end
        g{i+1,1}(j,1)=g{i,1}(j,1)+p{i}(j,1);%x 坐标
        g{i+1,1}(j,2)=g{i,1}(j,2)+p{i}(j,2);%y 坐标

[g{i+1,2}(j,2),g{i+1,2}(j,1)]=cart2pol(g{i+1,1}(j,1),g{i+1,1}(j,2));

```

```

end
%%
%计算误差
for j=1:size(dronecartesian,1)
    error(i,j)=linkCompute(g{i+1,1}(j,:),dronecartesian(j,:));
end
if sum(error(i,:))<1
    break;
else sum(error(i,:))>200
    break;
end
end
end

```