

无人机遂行编队飞行中的纯方位无源定位

摘要

纯方位无源定位,即由编队中某几架无人机发射信号、其余无人机被动接收信号,从中提取出方向信息进行定位,来调整无人机的位置。本文主要研究了圆形编队和锥形编队的无人机纯方位无源定位和调整模型。

针对问题一,首先讨论了圆形阵列中三架发出信号无人机的四种排布方式,并在极坐标下根据正弦定理分别讨论各无人机间的几何关系,将无人机的定位问题转化为方程问题。其次,求解方程得出无人机位置的一般解,建立了被动接受信号无人机定位模型,随后,为解决特殊情况下方程解的不唯一性,以及位于信号源连线角平分线上无人机位置的不确定性,提出了假设:已知信号源发射电磁波的次序。最后,给出该背景下无人机的位置调整策略:先沿圆的径向,再沿圆的切向进行调整,从而使模型更具完整性。

针对问题二,在无人机能够确定自身编号、实际位置与理想位置差别较小的假设条件下,利用接收到的角度确定发射信号无人机的编号。发现:无人机的实际位置与理想位置偏差量较小时,可以使用一架额外的无人机通过接收到的信号确定发射信号的无人机编号,进而实现对接收信号的无人机的有效定位。

针对问题三,对于初始位置略有偏差的圆形编队,首先对一般情况下无人机初始位置进行预处理:即将无人机 j 沿径向调整,使其从无人机 $0,1$ 两架无人机接收到的方位角调整为 $(90^\circ - 20^\circ(j - 1))$ 。其次通过数学证明,预处理后可以将无人机的距离误差自 15% 缩小至 5% 。随后,由于各无人机之间的信息不共享,故不能出现通过任何全局的信息对无人机做出调整。基于局部最优模型分两种方案进行调整策略的研究:每次选取圆周上三或两架无人机作为信号源。最后,对该模型进行仿真模拟,得出对于给定初始值的两种方案位置的调整结果。比较两个方案误差值的收敛速度、收敛值,最终得出方案二,即每次在圆周上选取两架无人机发射信号,更优。

针对问题四,将锥形编队分成四个全等三角形,依次对每个三角形中的六架无人机进行调整。利用问题三中确定的定位和调整模型调节三角形中的无人机,发现调整模型可行,位置总偏差小于 1m 。

关键词: 无源定位; 几何关系; 仿真模拟; 余弦定理; 遍历算法

一、问题重述

1.1 问题背景

无人机编队飞行中的纯方位无源定位问题是无人机技术中的一个重要研究领域，具有广泛的应用背景。在军事、民用和商业领域，无人机编队飞行已经展现出巨大的潜力和价值。随着无人机技术的不断发展和应用领域的不断扩大，纯方位无源定位技术将会得到更广泛的应用和进一步的研究。

纯方位无源定位利用无人机自身携带的传感器或设备，通过测量其他无人机或目标的方位信息，确定自身和其他无人机的位置和状态，从而实现精确的定位和导航。

1.2 问题要求

基于上述背景，要求建立数学模型解决如下问题：

问题的无人机阵型分为两类：

1) 圆形编队：由 10 架无人机组成，其中 9 架无人机(编号 FY01-FY09) 均匀分布在某圆周上，另 1 架无人机(编号 FY00) 位于圆心。

2) 锥形编队：由 15 架无人机组成(编号 FY01-FY15)，且直线上相邻两架无人机的间距相等。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。

问题一：圆形编队中，位于圆心的无人机和圆周上另 2 架无人机发射信号。发射信号的无人机位置无偏差且编号已知，被动接受信号的无人机位置略有偏差，建立被动接收信号的无人机定位模型。

问题二：圆形编队中，FY00 和 FY01 以及编队中若干编号未知的无人机发射信号。发射信号的无人机位置无偏差，被动接受信号的无人机位置略有偏差。除 FY00 和 FY01 外，还需要几架无人机发射信号才能实现被动接收信号无人机的有效定位。

问题三：圆形编队中，圆周半径为 100m，初始时刻无人机的位置略有偏差。根据题目数据，要求给出合理的无人机位置调整方案。即通过多次调整，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，使得 9 架无人机最终均匀分布在某个圆周上。

问题四：锥形编队中，考虑纯方位无源定位的情形，设计无人机位置调整方案。

二、问题分析

问题的研究对象是无人机集群，研究内容为保持编队队形的无人机位置确定和位置调整方案。该问题描述了通过部分无人机的确定位置以及接收到的方向信息，确定其他无人机的位置信息，并针对不同情况下的位置调整提出了不同的要求。

2.1 问题一分析

该问本质上是一个方程问题。由于已知位置的三架无人机位置无偏差，故可建立极坐标并运用正弦定理计算出被动接受信号无人机的位置。考虑到圆形阵列的特殊性，两架圆周上发出信号的无人机具有四种位置关系，故需对其进行分情况讨论。另外，注意到九等分圆的轴对称性，因此最终位置的解有两个，需假设接收信号无人机已知自身的编号。计算出待定位无人机的位置后，可根据圆内角、

圆周角、圆外角的性质给出其位置的调整方案,使之可以准确地消除位置偏差。

2.2 问题二分析

第二问要求给出实现定位的最小发射信号无人机架数,已知条件有无人机 FY00 与无人机 FY01 发射信号的角度,未知编号发射无人机与待测无人机的角度。由于待测无人机的位置的略微偏差会影响到角度的偏差,但偏差范围较小。由此,假定一架待测无人机(FY00 和 FY01 除外),我们对发射信号的未知编号无人机进行遍历,通过待测无人机与 FY00、编号未知无人机三者的夹角误差区间初步排除不可能的飞机编号,再通过待测无人机与 FY00、编号未知无人机三者的夹角误差区间确定发射信号飞机的编号,当飞机编号确定之后,问题就与第一问完全相同,可以进行精准定位。

2.3 问题三分析

该问本质上是一个局部优化问题。虽然无人机的初始位置与正九边形顶点有微小偏差,但我们可以通过一定手段(详见正文),称为预处理,将极径误差大幅下降。预处理后,分两种方案建立局部最优模型,即每次均匀选取圆周上两或三架无人机发射信号。为了评价两种方案的优劣性,可根据设置的误差函数比较两方案全局误差的收敛速度和收敛值,进而比较需要发射的次数。随后代入题目中已给的初始数据验证,并利用模拟生成初始位置的随机数来衡量模型的灵敏性。

2.4 问题四分析

本题要求实现锥形编队队形的无人机位置调整方案。我们假设编号 FY15 和 FY14 这两架无人机的位置是没有偏差的,而其他 13 架无人机的位置均有较小偏差。我们先将锥形无人机集群分为四个等边三角形(每个等边三角形集群中包涵 6 架无人机),通过对位置精确的两架无人机(FY14 和 FY15)所在的等边三角形的其他无人机进行层序遍历计算出精确位置并进行调整,再利用调整好的与其他等边三角形无人机集群相交的三架无人机进行相同操作确定另外 3 架无人机的准确位置并进行调整,最后对另两个等边三角形无人机集群的无人机进行调整,实现对该锥形无人机编队的纯方向无源定位。

三、模型假设

为建立更精确的数学模型,本文根据实际情况建立一些合理的假设及条件约束,具体假设如下:

3.1 问题一假设

假设一: 假设每架无人机都知道自己的编号;

假设二: 每架被动接受信号的无人机都会有一定的位置偏差,在极坐标表示下,假设每架无人机所处位置的偏差情况不超过 5% (包括极径与极角);

假设三: 假设无人机间信息不共享。接收信号的无人机是被动接受的,不能发射电磁波信息,故无人机之间的信息不共享,即无人机 i 不能利用其他无人机 j 接收到的方向信息(i 和 j 均是题目所给的无人机集群中的某一有确定编号的无人机)。

3.2 问题二假设

假设一: 假设无人机编队中的每架无人机与其对应的精确位置偏差较小。即不存在某一编号无人机与其编号对应的精确位置之间的偏差大于另一架无人机与该位置的偏差。

假设二: 相比于上一问,本问的关键在于如何确定发射信号的无人机编号,

需要接收信号的无人机自行判断。事实上,如果能确定发射信号的无人机的编号,就可以用上一问的模型进行求解。因此,根据待测飞机的实际位置与理想位置偏差大小,我们认为除了 FY00 和 FY01 之外,还需要一架发射信号的飞机,才能实现飞机的有效定位。

3.3 问题三假设

假设一: 无人机不可以沿某一方向移动指定距离;在对问题三模型进行求解时,设置搜索邻域、搜索步长是为了对无人机实际飞行情况进行仿真模拟。实际情况是无人机在原来位置附近任意移动,搜索目标函数最小值。

假设二: 当误差小于一定值时,可以认为无人机以正九边形遂行编队飞行;考虑到实际情况,无人机群构成严格正九边形是不可能的,总会有一定的误差。所以我们认为,当无人机群构成的九边形与正九边形误差小于一定值时,即认为无人机群构成正九边形。

3.4 问题四假设

假设一: 假设无人机的位置调整可以沿任意方向。现实中的无人机飞行方向与设计性能相关,一般无人机在空中拥有 12 个自由度,为了迎合我们建立的模型,我们假设无人机在调整位置时可沿任何方向,以此简化模型及计算的复杂度。

假设二: 假设当误差小于某一阈值时,即可判定该无人机已经达到精准定位。考虑到实际情况,无人机集群难以构成位置绝对准确的编队集群,总会有一定的误差,而当误差足够小时,我们认为这一架无人机实现了精准定位。

四、符号说明

符号	含义
θ	极轴与原点 and 另一架主动机的连线的夹角
φ	极轴与原点 and 被动接收信号无人机(被动机)的连线夹角
α_i	被动机关于不同主动机的方位角(i 的取值所代表的无人机编号在下文具体给出)
(ρ_i, θ_i)	极坐标下无人机 i 的坐标点
ε_k	编号为 FY0k 的无人机的半径误差
ξ_k	编号为 FY0k 的无人机的角度误差
γ_{ij}	锥形无人机队列中以接收信号无人机为中心的 正方形点阵中第 i 行 j 列位置与该无人机理想位置的偏差值

注: 其他符号将在下文给出详细说明。

五、模型的建立和求解

5.1 问题一: 被动接受信号无人机的定位调整模型

5.1.1 模型的建立

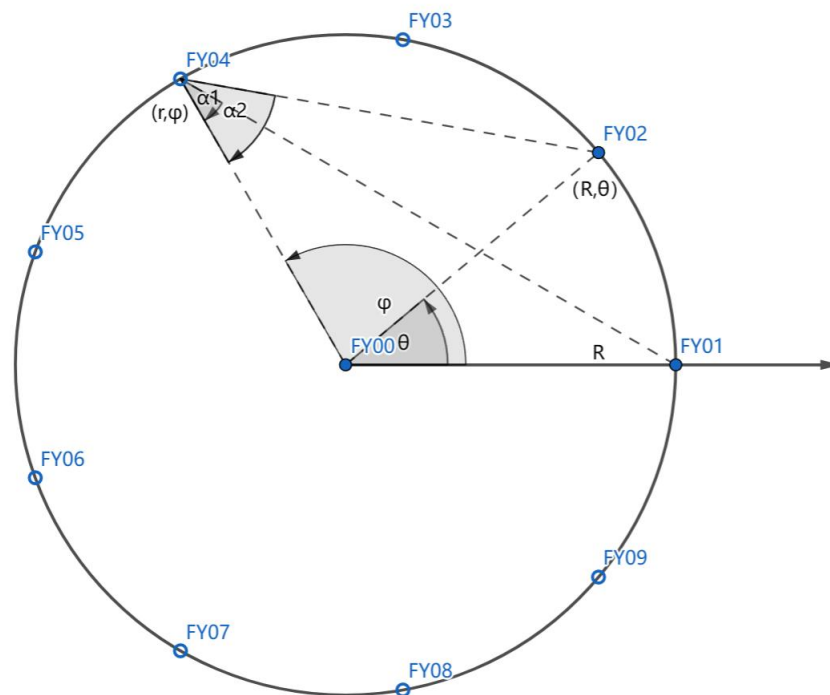
由于无人机的分布呈辐射对称的状态,因此选择圆周上第一架主动发出信号的无人机(主动机)是任意的,我们可以将其标记为 FY01。以 FY00 为原点,建立极坐标系,其中 FY00 与 FY01 连线方向为极轴,逆时针为正方向。

定义变量如下:

- 极轴与原点 and 另一架主动机的连线夹角为 θ

- 原点与被动接受信号无人机（被动机）的连线夹角为 ϕ
- 主动机的次序用编号大小来表示，编号越小次序越低
- 被动机关于前两架主动机的方位角为 α_1
- 被动机关于第一、第三两架主动机的方位角为 α_2
- 被动机关于第二、第三两架主动机的方位角为 α_3
- 圆周上两架主动机的位置分别为 $(R, 0)$ 和 (R, θ) ，被动机的位置为 (r, ϕ)

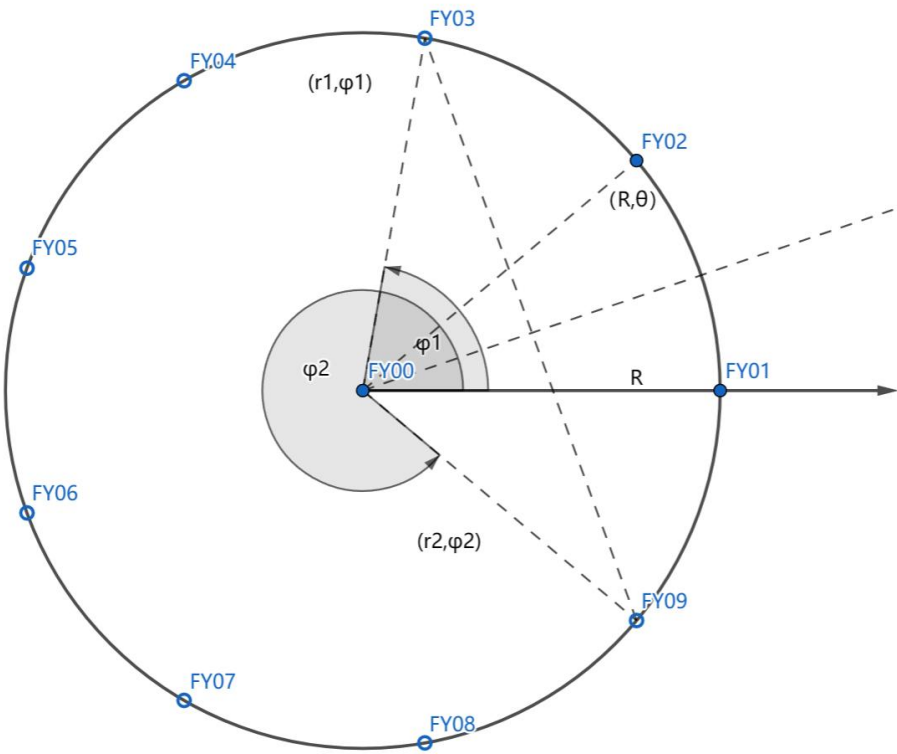
由于主动机位置无偏差，被动机位置略有偏差，故设圆周上两架主动机的位置分为 $(R, 0)$ ， (R, θ) ，被动机的位置为 (r, ϕ) 。主动机与被动机的排布具有两种情况： $\theta > \phi$ 和 $\theta < \phi$ 。



θ 与 φ 的取值范围不会影响数值解的大小,因此,我们可以只讨论 θ 与 φ 的大小关系。以 $\theta \in [0, \pi) \cap \varphi \in [0, \pi)$ 为例,设两架第二架主动机分为FY02, FY04; 两架被动机分为FY04, FY02, 具体图示如上图所示。

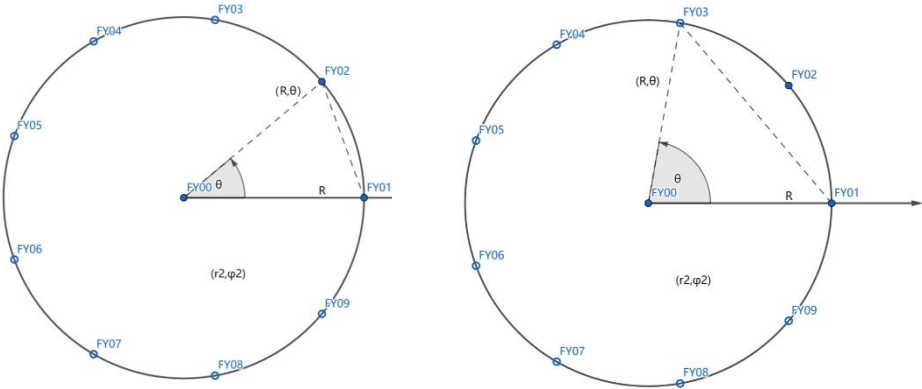
这两组解对应的被动机位置以三架主动机对应圆心角的角平分线为轴对称分布,如下图所示。特殊地,当被动机与相应角平分线的延长线重合时,方程的两个解相等。

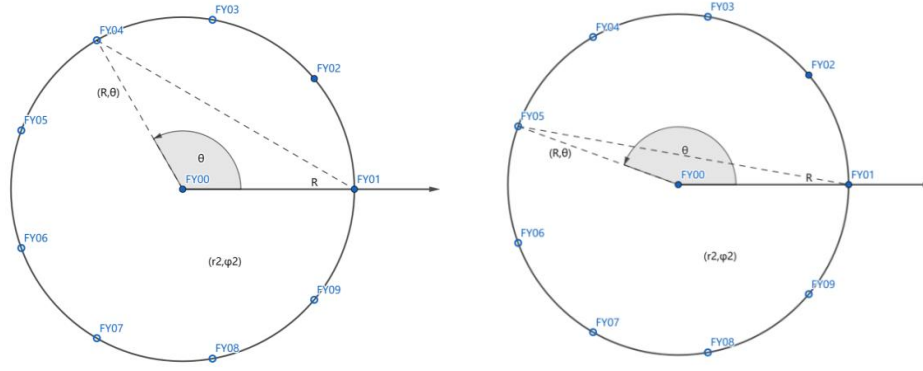
为了准确知晓被动机的具体位置,我们提出假设:被动接收信号的无人机实际已知自己对应的编号。基于此假设,可以确定被动机位置的唯一解。在实际情况中,一定编号的无人机对应的遥控设备往往固定,故此假设是合理的。



5.1.3 被动接收信号无人机位置调整方案

在上一步定位被动机后可更明确地知道其偏离程度。为了减小被动机的偏离程度,我们提出了被动机位置的调整方案。由于圆的高度对称性,主动机的排布共有四种情况,如下图所示:





以排布方式 1 为例讨论被动机位置的调整方案, 其余三种排布同理。如图所示, $FY00, FY01, FY02$ 为主动机, P 点为被动机所在位置, 由 5.1.1 中方位角的定义知, $\angle F0PF1 = \alpha_1$, $\angle F0PF2 = \alpha_2$, $\angle F1PF2 = \alpha_3$, 易知 $\angle F1F0F2 = 40^\circ$ 。

若 P 的位置无偏差, 则各无人机对应的方向角如表 1 所示。

		FY04	FY05	FY06	FY07	FY08	FY09
	FY03						
α_1	50°	30°	10°	10°	10°	30°	50°
α_2	70°	50°	30°	20°	30°	50°	70°
α_3	20°	20°	20°	20°	20°	20°	20°

若 P 的位置有偏差, 则可将被动机的位置分为两种情况讨论。

1. 当 P 点在圆周上时, 方位角满足:

$$\begin{cases} \alpha_1 = 90^\circ - \frac{\varphi}{2} \\ \alpha_2 = 90^\circ - \frac{\varphi - 40^\circ}{2} \\ \alpha_3 = 20^\circ \end{cases}$$

2. 当 P 点不在圆周上时, 根据圆的性质: 同弧所对的圆外角 < 圆周角 < 圆内角, 则对于 $\widehat{F1F2}$, P 在圆外时, $\alpha_3 < 20^\circ$; P 在圆内时, $\alpha_3 > 20^\circ$ 。

因此, 若要将有偏差的被动机调至无偏差, 应按以下步骤进行:

Step1: 观测被动机接收到的方位角信号, 并利用 α_1 与 α_2 计算出 α_3 ;

Step2: 比较 α_3 与 20° 的大小, 若 $\alpha_3 > 20^\circ$, 执行 Step3; 若 $\alpha_3 < 20^\circ$, 执行 Step4; 若 $\alpha_3 = 20^\circ$, 执行 Step5;

Step3: 若被动机在圆内, 应沿径向背离圆心飞行, 直至 $\alpha_3 = 20^\circ$;

Step4: 若被动机在圆外, 应沿径向朝向圆心飞行, 直至 $\alpha_3 = 20^\circ$;

Step5: 若被动机在圆周上, 进行角度微调。应沿切线方向飞行(无人机不施加朝向圆心的力), 直至 α_1 与 α_2 和无偏差的方向角重合。

5.2 问题二: 无源定位的无人机数量最小需求

5.2.1 误差较小的情况

根据圆的几何知识, 我们可以知道, 在理想状态下, 待测飞机与 $FY00$ 、 $FY01$ 形成待测飞机为顶角的角度可能存在 10° , 30° , 50° , 70° 四种情况(如图 2

所示)；待测飞机、FY01、另一架位于圆周上理想位置的飞机形成的以待测飞机为顶角的角度可能存在 20° ， 40° ， 60° ， 80° ， 100° ， 120° ， 140° 几种情况（如图 3 所示）。

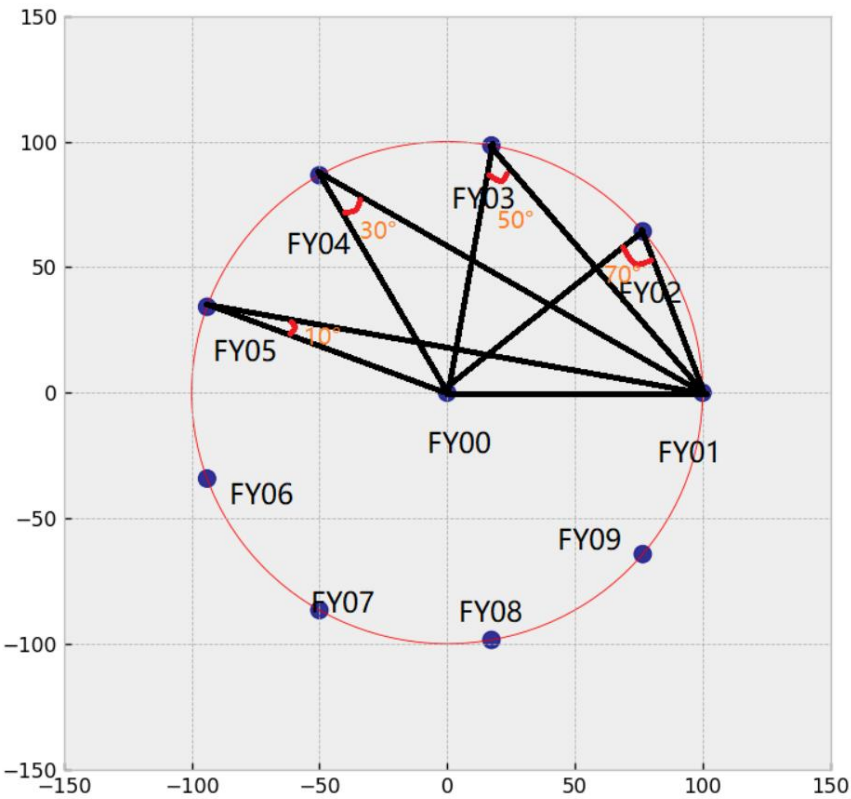


图 2：待测飞机与 FY00、FY01 的理想位置角度

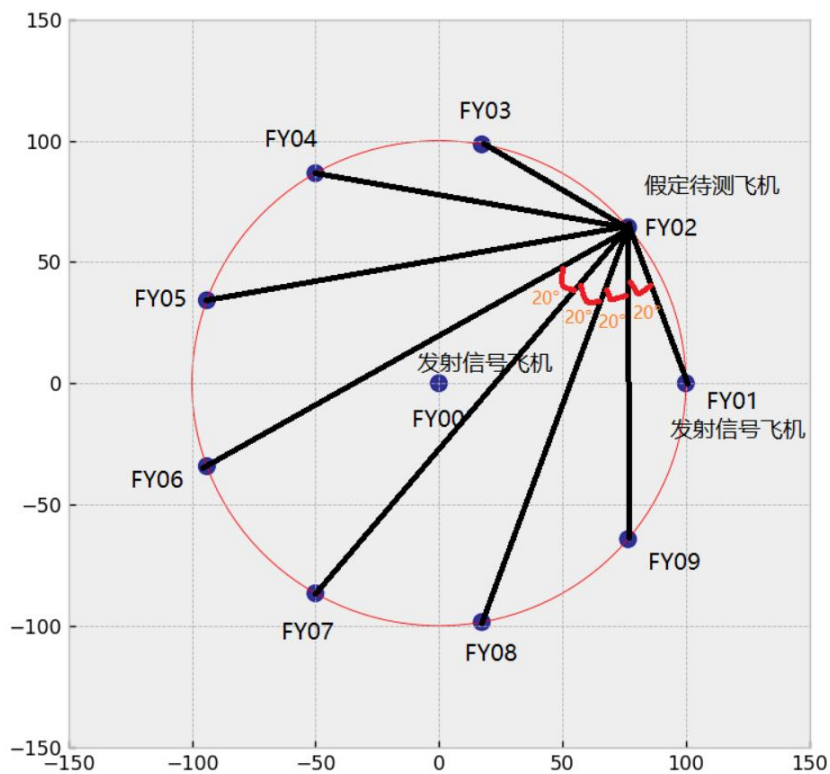


图 3：待测飞机与 FY01、未知发射信号飞机的理想位置角度

Step1：确定另一发射信号飞机的编号

当待测飞机实际位置与理想位置有较小偏差时，可以先根据未知编号发射信号飞机与接收信号飞机、FY00 的顶角（以接受信号飞机为定点）确定编号未知飞机两个可能的位置编号，例如：以 FY02 为待测飞机，未知编号发射信号飞机与 FY02、FY00 形成的夹角大约为 50° 时，未知编号发射信号飞机可能为 FY04，也可能为 FY09。

确定了未知编号发射信号飞机的基本位置可能情况下，再根据待测飞机与未知编号发射信号飞机、FY01 的顶角（待测飞机为顶角）区分，例如：以 FY02 为待测飞机，实际情况下，FY02 与 FY04、FY01 形成的顶角（FY02 为顶角）大约为 120° ；而 FY02 与 FY09、FY01 形成的顶角（FY02 为顶角）大约为 20° ，因此可以根据待测飞机接收到的角度信息区别 FY04 和 FY09。由此确定未知编号发射信号飞机编码。

Step2：构建模型确定待测飞机实际位置

通过 step1 可以准确知道未知编号发射信号飞机的编号，又已知 FY00、FY01，三者都处于理想准确位置，可以通过第一问建立的双圆定位模型确定待测飞机的实际位置。

5.2.2 误差较大的情况

当实际位置与理想位置相差较大时，通过计算机多次计算测试表明：如果只有三架发射信号飞机，很难确定待测飞机的实际位置，还需要再增设一架发射信号飞机才能确定待测飞机的实际位置。

5.3 问题三：

5.3 圆形编队的具体调整方案

根据题干信息, 可将此问题分为四个步骤进行。首先基于一般的模型, 对无人机初始位置进行预处理, 证明通过无人机径向的调整后, 可使得距离误差缩小。其次, 分两种方案建立局部最优模型, 即每次选取圆周上两或三架无人机发射信号。随后, 以两个方案中每个点最后一次调整后对应目标函数的和作为误差函数, 来评估无人机的调整方案。最后代入题目中已给的初始数据, 并利用计算机模拟生成初始位置的随机数来衡量模型的灵敏性。

5.3.1 无人机初始位置的预处理

记理想无人机所围成圆的半径为 R 。由于图形的相似性(两个相似的图形, 对应角相等), 故必须有一架无人机距圆心距离为 R 。记极坐标下无人机位置分别为 $(0, 0^\circ), (R, 0^\circ), (R + \varepsilon_k, 40^\circ(k - 1) + \xi_k), k = 2, 3, \dots, 9$, 其中 ε_k 与 ξ_k 分别为半径

误差与角度误差。由于初始时刻无人机的位置略有偏差, 根据题干中表 1 给出的数据, 不妨设 $\varepsilon_k \in (-15\%R, 15\%R)$, $\xi_k \in \{-0.5^\circ, 0.5^\circ\}, k = 2, 3, \dots, 9$

对初始时刻无人机的位置进行预处理, 使得对 $\forall k \in \{2, 3, \dots, 9\}$, 距离误差 ε_k 可缩小至 $(-5\%R, 5\%R)$, 预处理过程如下:

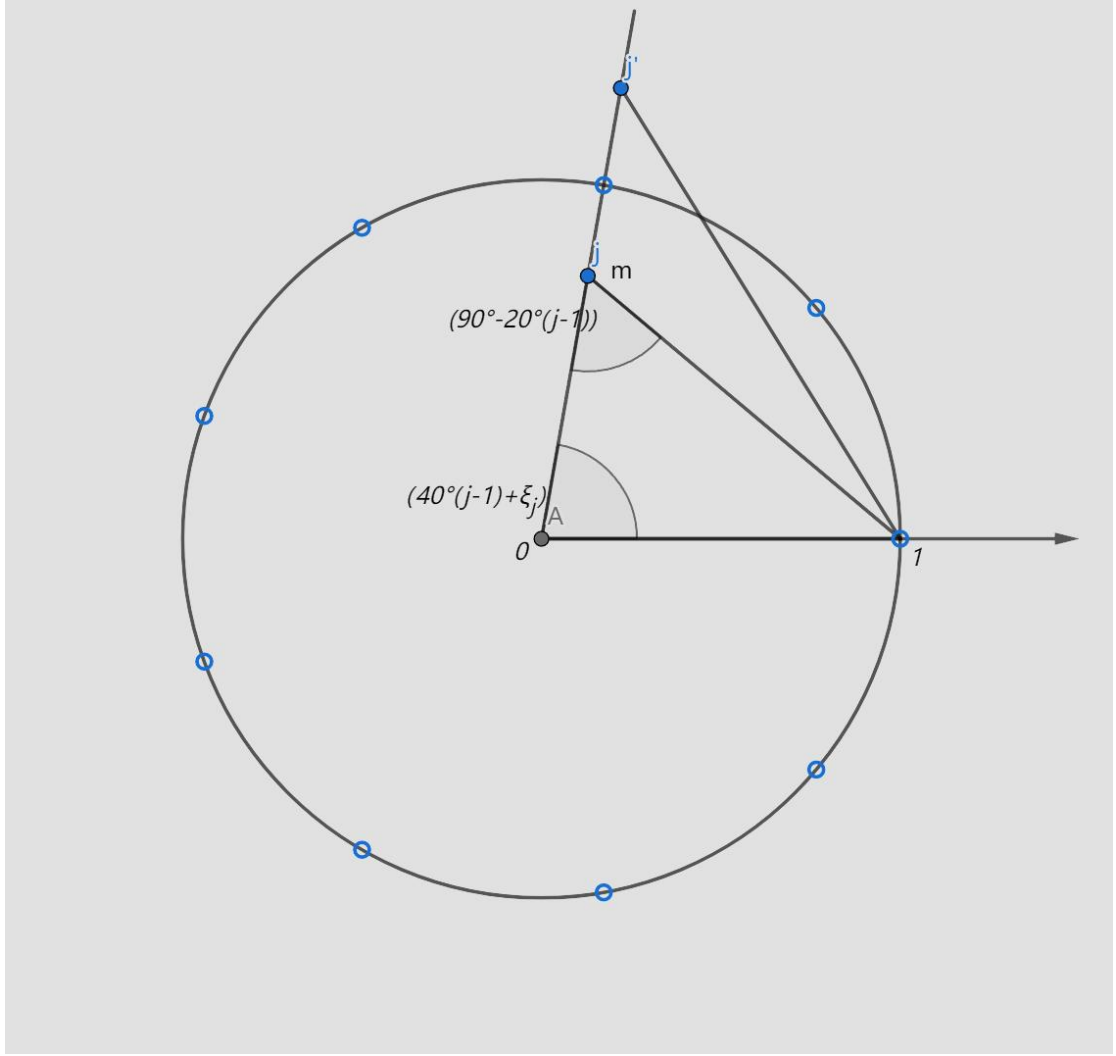
- 对任意无人机 $j \in \{2, 3, \dots, 9\}$, 其极坐标为 $(R + \varepsilon_j, 40^\circ(j - 1) + \xi_j)$. 将其沿径

向调整, 使其从无人机 0, 1 接收到的方向信息角调整为 $\alpha_j = 90^\circ - 20^\circ(j - 1)$ 。

- 沿径向调整是因为, 无人机 j 只知道来自 0, 1 无人机的方向信息角, 不知道其极

角为 $40^\circ j + \xi_j$, 沿其他方向调整很可能会使其极角更大程度的偏离 $40^\circ j$

下证通过对无人机初始位置预处理, 可以将距离误差缩小至 $\varepsilon_k \in (-5\%R, 5\%R)$:



预处理证明示意图

由正弦定理可得

$$\frac{0j'}{\sin [180^{\circ} - (90^{\circ} - 20^{\circ}(j-1)) - (40^{\circ}(j-1) + \xi_j)]} = \frac{R}{\sin [90^{\circ} - 20^{\circ}(j-1)]}$$

整理得

$$\begin{aligned} \frac{0j'}{R} &= \frac{\cos [20^{\circ}(j-1) + \xi_j]}{\cos [20^{\circ}(j-1)]} = \cos (\xi_j) - \tan (20^{\circ}k) \sin (\xi_j) \\ &\leq \cos (-0.5^{\circ}) - \tan (20^{\circ} * 4) \sin (-0.5^{\circ}) \\ &\approx 1.049 \end{aligned}$$

从而距离误差

$$\varepsilon_j = \frac{|R - 0j'|}{R} \approx 0.049 < 5\%$$

由此可见,对无人机初始位置预处理是必要的。一方面,预处理可使距离误差缩小至更小的范围,更重要的是,通过下面局部最优模型可以看出,初始位置预处理将大大减小目标函数最优值搜索次数。我们知道,一次搜索对应 k 次无人机向外发射电磁波信号。初始位置预处理意味着将大大减少无人机向外发射电磁波信号

的次数。

5.3.2 局部最优模型的建立

由于无人机定位系统的无源性，各无人机之间不共享信息，因此无法通过全局调整无人机位置。

我们将通过局部最优模型，实现无人机位置尽可能均匀分布在以 R 为半径的圆周，同时使无人机向外发射电磁波信号的总次数尽可能少。

以无人机 0 为原点，01 方向为 x 轴正方向建立平面直角坐标系 xOy ，并假设 1 的位置 $(R, 0)$ 是准确的。

依据题干信息，下面介绍两种调整方案，一种是每轮调整仅选取圆周上三架无人机发射信号，其余无人机被动接收信号并作出调整；另一种是每轮调整仅选取圆周上两架无人机发射信号，其余无人机被动接受信号并做出调整。最后，定义误差函数，量化两种方案调整的好坏。

方案一：每次选取圆周上三个无人机发射信号

考虑到正九边形的对称性，258、369、471 三组无人机均近似构成等边三角形，这些无人机依次作为发射信号的信号源，其余无人机（除无人机 1 以外）接收信号根据局部最优模型做出调整。注意到无人机 1 的位置是准确的，因此始终不调整。

Step1: 选取 0258 为发射信号的信号源，134679 无人机被动接受信号并作出调整（1 不需要调整）。由于圆的对称性，以无人机 3 为例（其余无人机与 3 无本质区别），其调整方式如下：

记 $\angle 530 = \alpha, \angle 038 = \beta, \angle 032 = \gamma$ 。由正九边形几何关系易知， α, β, γ 三个角的理想值分别为 $50^\circ, 10^\circ, 70^\circ$ 。建立目标函数

$$f_3(\alpha, \beta, \gamma) = (\alpha - 50^\circ)^2 + (\beta - 10^\circ)^2 + (\gamma - 70^\circ)^2$$

在无人机编号为 f_3 的情况下，其余无人机的目标函数为 f_4, f_6, f_7, f_9 。以无人机 3 为中心，在边长为 a 的正方形区域内以步长 b 进行搜索，以使目标函数最小化。实际情况中，无人机可以在其原始位置附近进行任意搜索，而边长和步长的设置仅用于仿真模拟无人机调整过程。

Step2: 选取 0369 为发射信号的信号源，124578 无人机被动接受信号并作出调整（1 不需要调整）。调整方式同 **Step1**，无人机 2 的目标函数记为 g_2 。其余待调整无人机目标函数为 g_4, g_5, g_7, g_8 。

Step3: 选取 0471 为发射信号的信号源，235689 无人机被动接受信号并作出调整（1 不需要调整）。调整方式同 **Step1**，无人机 2 的目标函数记为 h_2 。其余待调整无人机目标函数为 h_3, h_5, h_6, h_8, h_9 。

Step4: 设置循环次数阈值 η ，以控制无人机调整结束。设总循环次数为 σ ，我们对一次循环如下定义：选取 0258 为发射信号的信号源，其他无人机作出调整，然后选取 0369 为发射信号的信号源，其他无人机作出调整，最后选取 0471 为发射信号的信号源，其他无人机作出调整。一次循环每架无人机（无人机 0,1 除外）均被选做一次发射信号的信号源与选做两次被动接受信号无人机做出调整。也就是说，若总循环次数为 σ ，各无人机调整次数如表 2 所示：

飞机编号	0	1	2~9
调整次数	0	0	2σ

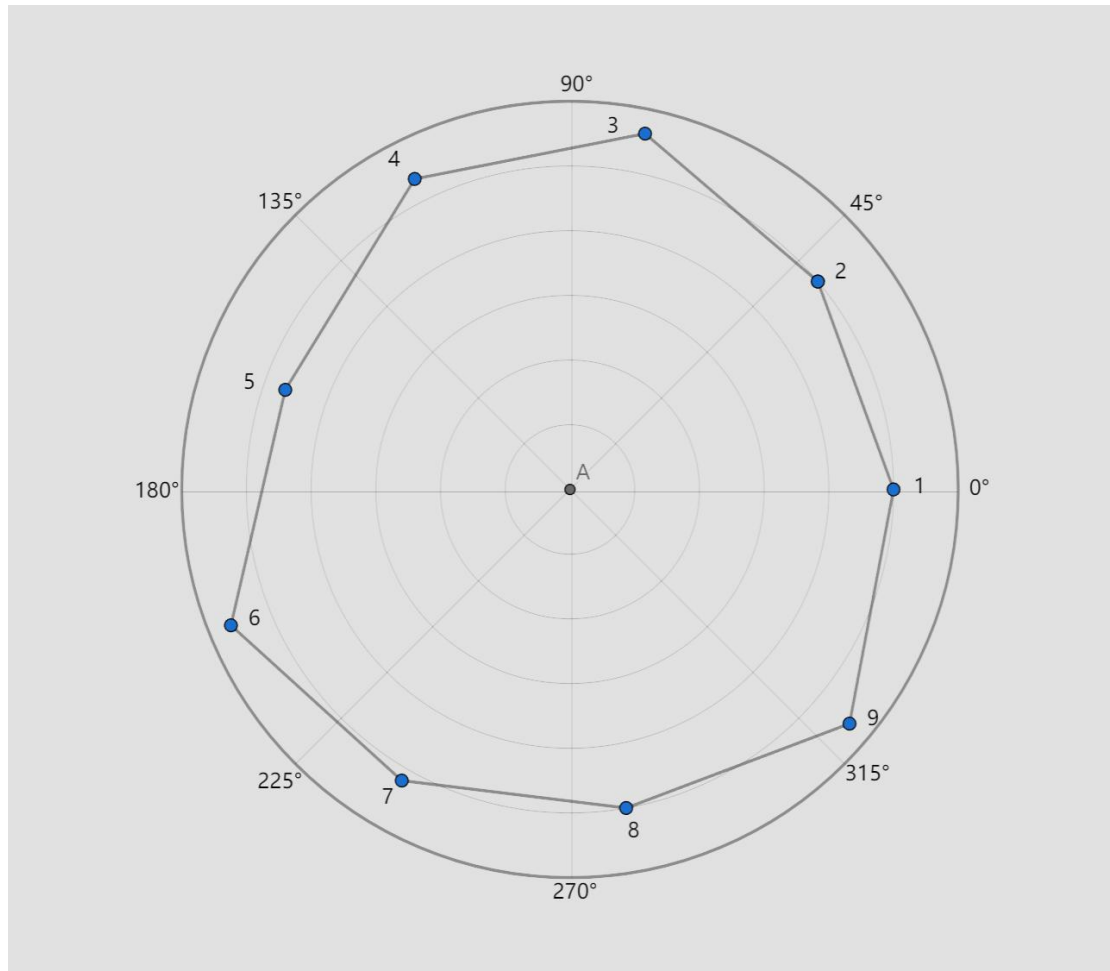
若每架无人机（0,1 除外）调整次数超过 η ，即 $\sigma > \frac{\eta}{2}$ ，则该飞机调整结束。否则， $\sigma \leq$

$\frac{\eta}{2}$, 返回 **Step1** 继续调整。

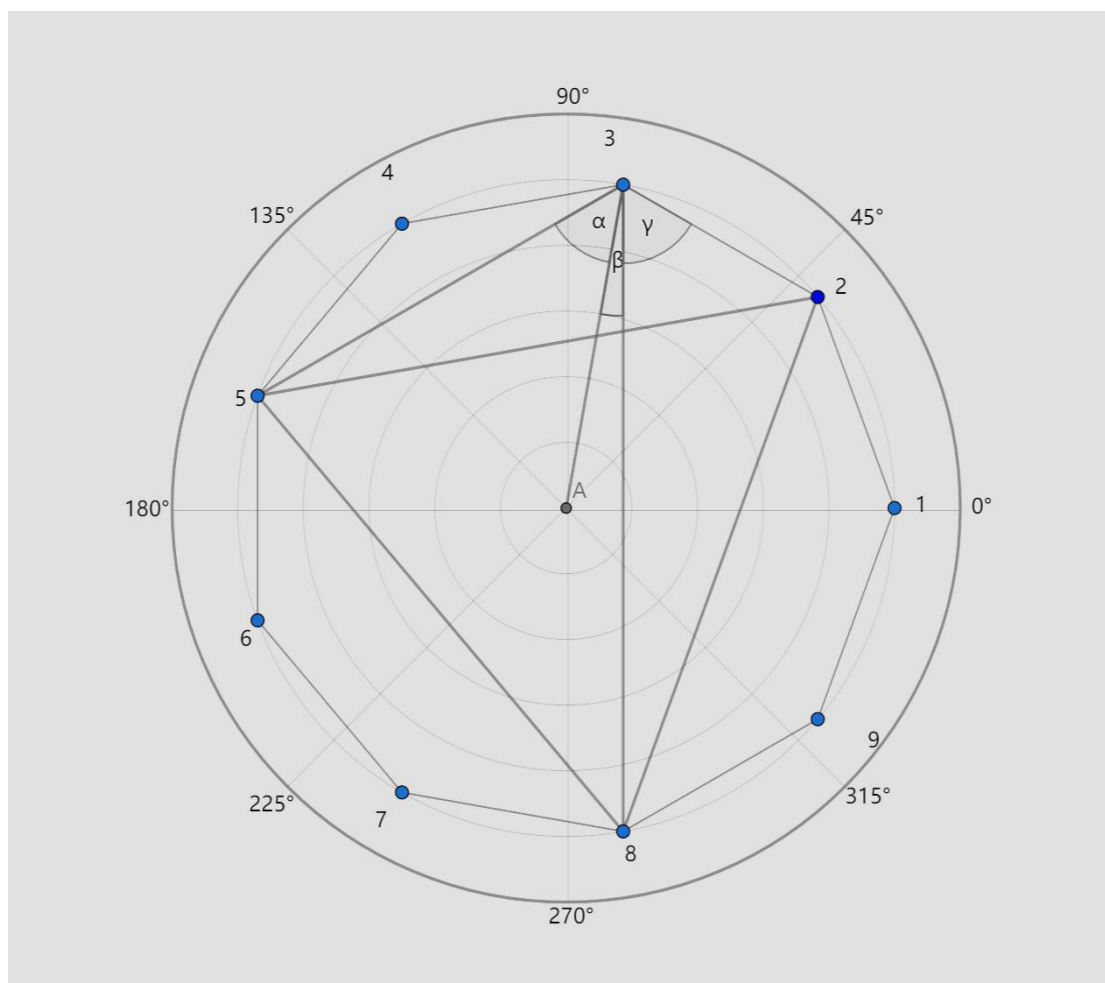
Step5: 定义误差检测函数

$$\mu = h_2 + h_3 + g_4 + h_5 + h_6 + g_7 + h_8 + h_9$$

量化无人机趋于正九边形程度。



方案一预处理前无人机的排布



方案一预处理后无人机的排布

方案二:每次选取圆周上两个无人机发射信号

分别选取 027、038、049、015、016 无人机发射信号, 调整方案与方案一本质相同, 仅以 025 无人机发射信号为例。

Step1: 选取 027 为发射信号的信号源, 1345689 无人机被动接受信号并作出调整 (1 不需要调整)。以无人机 9 接收无人机信号为例 (其余无人机与 3 类似, 目标函数见表 3), 其调整方式如下:

记 $\angle 790 = \alpha$, $\angle 092 = \beta$ 。由正九边形几何关系, 知 α, β 两个角的理想值均为 50° 。建立目标函数

$$f_9(\alpha, \beta) = (\alpha - 50^\circ)^2 + (\beta - 50^\circ)^2$$

f_9 下标为对应无人机编号。依此, 其余待调整无人机目标函数为 f_3, f_4, f_5, f_6, f_8 。以无人机 9 为中心, 在边长为 a 的正方形区域内以步长 b 调整, 使目标函数达到最小值。

表 3 目标函数序列

	2	3	4	5
027		$(70^\circ, 10^\circ)$	$(30^\circ, 50^\circ)$	$(30^\circ, 50^\circ)$
038	$(70^\circ, 30^\circ)$		$(70^\circ, 10^\circ)$	$(50^\circ, 30^\circ)$
049	$(50^\circ, 50^\circ)$	$(70^\circ, 30^\circ)$		$(70^\circ, 10^\circ)$
015	$(70^\circ, 30^\circ)$	$(50^\circ, 50^\circ)$	$(70^\circ, 30^\circ)$	

016	(70° , 10°)	(50° , 30°)	(50° , 30°)	(70° , 10°)
-----	--------------	--------------	--------------	--------------

续表

	6	7	8	9
027	(70° , 10°)		(70° , 30°)	(50° , 50°)
038	(50° , 30°)	(70° , 10°)		(70° , 30°)
049	(50° , 30°)	(50° , 30°)	(70° , 10°)	
015	(70° , 10°)	(50° , 30°)	(50° , 30°)	(70° , 10°)
016		(70° , 30°)	(50° , 50°)	(70° , 30°)

Step2: 选取 038 三架为发射信号的信号源, 无人机 1245679 被动接受信号并作出调整(1 不需要调整)。调整方式同 **Step1**, 目标函数记为 g_i 。

Step3: 选取 049 三架为发射信号的信号源, 无人机 1235678 被动接受信号并作出调整(1 不需要调整)。调整方式同 **Step1**, 目标函数记为 h_i 。

Step4: 选取 015 三架为发射信号的信号源, 无人机 2346789 被动接受信号并作出调整。调整方式同**Step1**, 目标函数记为 u_i 。

Step5: 选取 016 三架为发射信号的信号源, 无人机 2345789 被动接受信号并作出调整。调整方式同 **Step1**, 目标函数记为 v_i 。

Step6: 设置循环次数阈值 η 来控制调整结束。仍记总循环次数为 σ 。 σ 参数含义同上。各无人机调整次数如下表:

表 4 步骤二循环时各无人机的调整次数

飞机编号	0	1	2~9
调整次数	0	0	4 σ

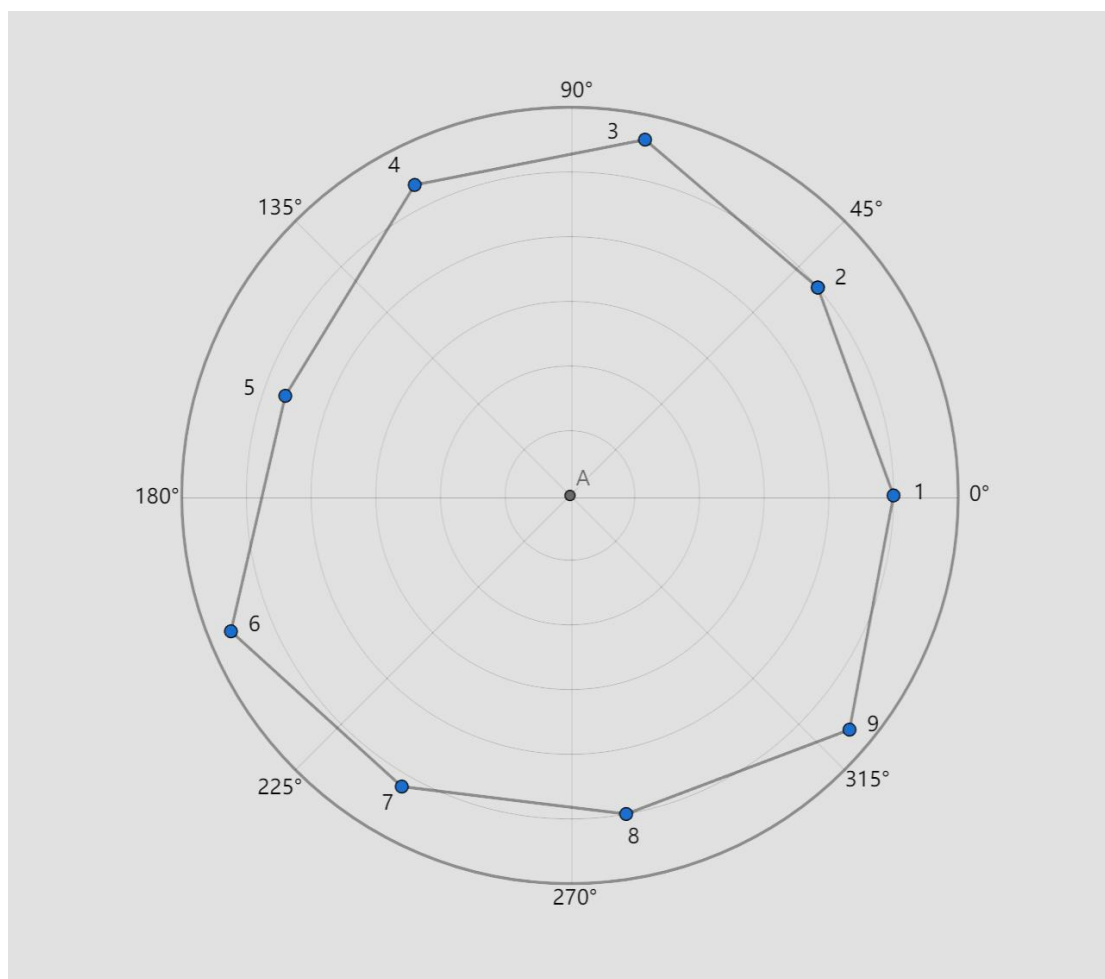
若每架无人机 (0,1 除外) 调整次数超过 η , 即 $\sigma > \frac{\eta}{4}$, 则该飞机调整结束。否则, $\sigma \leq$

$\frac{\eta}{4}$, 返回 **Step1** 继续调整。

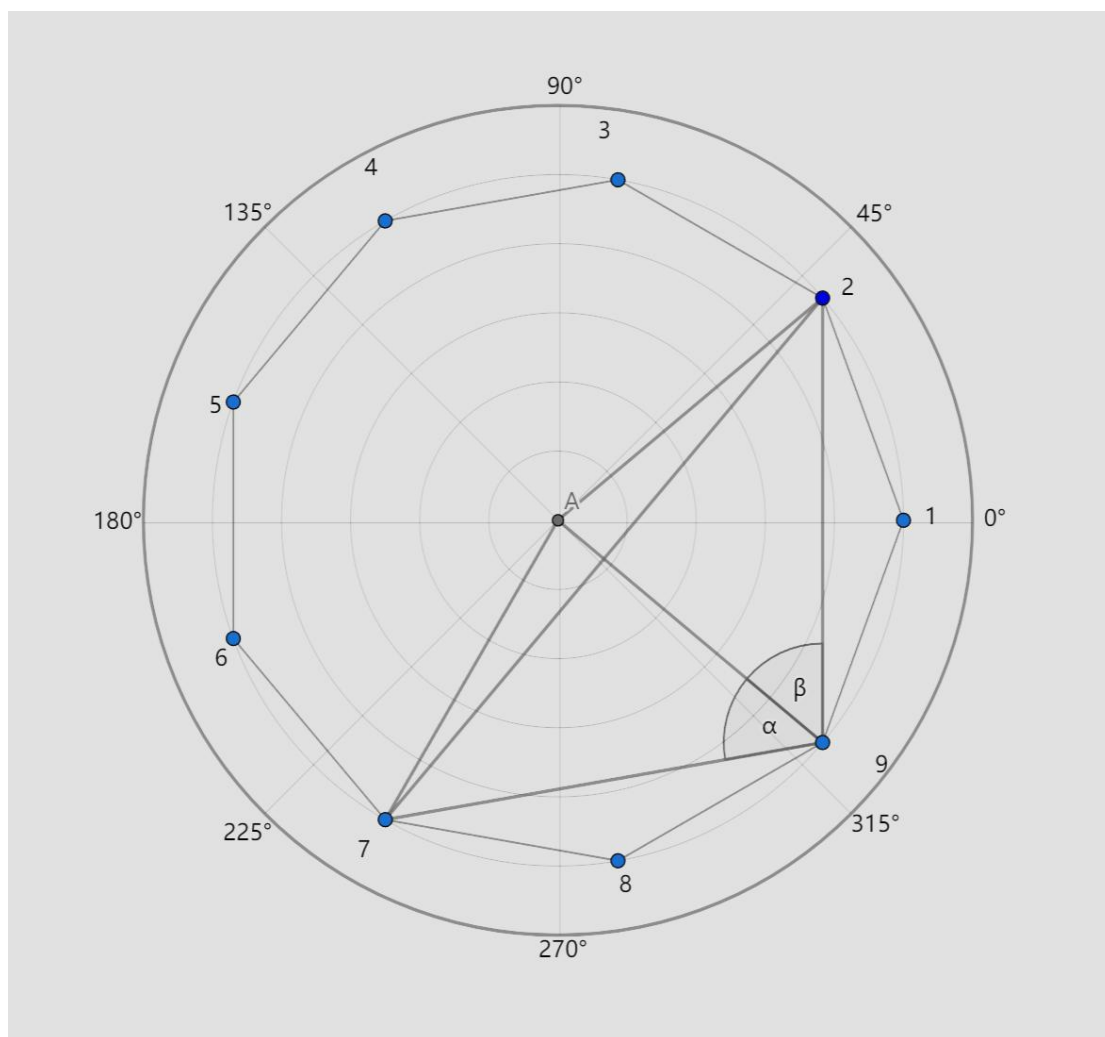
Step7: 定义误差检测函数

$$\mu = v_2 + v_3 + v_4 + v_5 + u_6 + v_7 + v_8 + v_9$$

量化无人机趋于正九边形程度。



方案二预处理前无人机的排布



方案二预处理后无人机的排布

5.3.3 局部最优模型的检验

两种方案全局检验误差关于总循环次的具体值见下表：

表 5 两种方案的误差关于总循环次数的具体值

次数	1	2	3	4	5	6	7	8
方案一	0.0969	1.90e-06	1.24e-06	8.36e-07	7.26e-06	5.96e-06	5.96e-06	5.96e-06
方案二	0.0956	0.0523	0.0446	0.0302	0.0254	0.0160	0.0067	0.0009

续表

次数	9	10	11	12	13	14	15	16
方案一	5.96e-06	5.96e-06	5.96e-06	5.96e-06	5.96e-06	5.96e-06	5.96e-06	5.96e-06
方案二	0.0002	5.35e-05	5.35e-05	5.35e-05	5.35e-05	1.01e-05	7.42e-05	7.42e-05

对比两个方案可以得出如下两点结论：

1) 方案二的收敛速度要明显优于方案一的收敛速度

方案二在循环工次后，全局检验误差就从 0.1 降到了 $1.901e-06$ ，而方案二需要循环 7 次才能进全局检验误差从 0.1 缩小至 0.0067 即方案二所有无人机调整 $64\sigma = 64$ 次，全局检验误差就可以缩小至很小的收敛值：而即方案一所有无人机通过调整 $48\sigma = 48 \times 7 = 336$ 次，全局检验误差才可以缩小至较小的收敛值，收敛值约为前者十倍。

2) 方案二的全局检验误差的收敛值远远小于方案一的全局检验误差的收敛值

方案二的全局检验误差在循环 6 次后达到稳定值 $5.961e - 06$, 方案二的全局检验误差循环 15 次才达到稳定值 $7.42e - 05$ 。

综上所述, 方案二, 即每次调整选取圆周上两个无人机发射信号, 比方案一更优。

5.4 问题四:

5.4.1 模型的建立

5.4.1.1 基准点的选取

本文以 FY15、FY14 与 FY13 作为基准点, 然后以 FY15 与 FY14 所在直线为 Y 轴, 以垂直于 Y 轴的直线为 X 轴。 但该模型支持以 FY01、FY11、FY15 向两边延伸一个无人机进而形成如 FY15 与 FY14 这样的组合作为基准点, 以该组合所在直线为 Y 轴, 以垂直于 Y 轴的直线为 X 轴, 进而通过我们建立的模型调整其他无人机的位置, 达到预期调整结果。

5.4.2 调整方案设计

5.4.2.1 调整方案步骤一

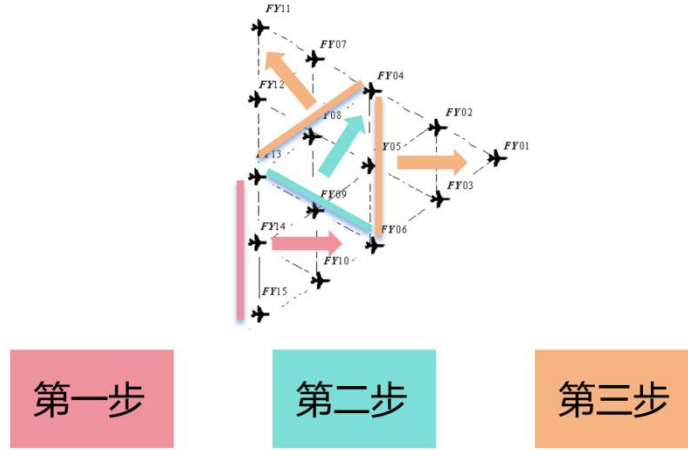
首先计算出 FY01 至 FY15 的理想坐标, 得到如下直角坐标:

无人机序号	坐标
FY01	(173. 20508075688772, 100)
FY02	(129. 9038105676658, 125)
FY03	(129. 9038105676658, 75)
FY04	(86. 60254037844386, 150)
FY05	(86. 60254037844386, 100)
FY06	(86. 60254037844386, 50)
FY07	(43. 30127018922193, 175)
FY08	(43. 30127018922193, 125)
FY09	(43. 30127018922193, 75)
FY10	(43. 30127018922193, 25)
FY11	(0, 200)
FY12	(0, 150)
FY13	(0, 100)
FY14	(0, 50)
FY15	(0, 0)
以上坐标分别对应 FY01 至 FY15。	

5.4.2.2 调整方案步骤二

本问题通过先调整一部分无人机的实际位置, 使其接近于理想位置, 然后再通过调整过的无人机调整剩下的无人机。下面以 FY06 为例演示如何根据 FY13、FY14、FY15 获得其相对理想的坐标:

首先在 FY06 理想坐标附近圈定一定区域 (本例以正方形为例), 然后将此区域进行细分, 我们选择以一边长为 32 米的正方形为圈定区域, 因为 FY03 无人机实际位置与理想位置略有偏差, 所以此正方形足以覆盖 FY03 无人机的实际位置, 然后将我们此正方形进行 800×800 等分 (当然只要计算机计算能力足够强大, 理论上可以无限等分), 形成点阵。



锥形无人机集群纯方向无源定位步骤图

接下来，先计算 FY06 与 FY13，FY14；FY06 与 FY14，FY15 的理想角度（均以 FY03 为顶点坐标），假定为 α 与 β 。

再通过层序遍历法，计算上述点阵上每一点与 FY13 和 FY14 以及 FY14 和 FY15 夹角的角度，假定为 $\alpha_i (i = 1, 2, \dots, 800)$ 与 $\beta_j (j = 1, 2, \dots, 800)$ ，并计算其与理想角度的偏差函数值 γ_{ij} ，通过比较得出 γ_{ij} 的最小值，进而找出与 α 和 β 最为接近的 α_i 和 β_j 。下面列出计算 γ_{ij} 值的通用模型：

$$\cos \alpha_i = \frac{b^2 + c^2 - a^2}{2bc}$$

其中 b 为遍历点与 FY13 的距离，c 为遍历点与 FY14 的距离，a 为 FY13 与 FY14 的距离；

$$\cos \beta_j = \frac{b^2 + c^2 - a^2}{2bc}$$

其中 b 为遍历点与 FY15 的距离，c 为遍历点与 FY14 的距离，a 为 FY15 与 FY14 的距离；

进而运用下列公式计算出偏差函数 γ_{ij} 的值：

$$\gamma_{ij} = (\alpha - \alpha_i)^2 + (\beta - \beta_j)^2$$

将得到的所有 γ_{ij} 的值进行比较，取 γ 最小值的 i 与 j，将 (i, j) 视为 FY06 的精准位置。

调整 FY06 的实际坐标后，再根据 FY06 的坐标调整 FY13 的坐标，并如上再次进行如上计算与调整（即迭代法），位置的准确性随计算次数的增加而提高。

接着运用 FY13、FY14、FY15 (这三架无人机经过上述调整已经可以视为与理想位置没有偏差) 作为基准点调整 FY09、FY10 两架无人机的位置。

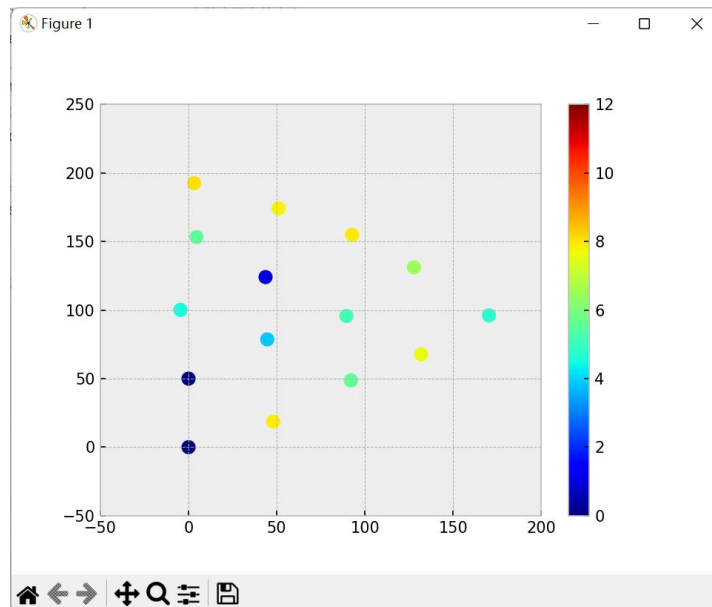
进而以 FY13、FY09 与 FY06 无人机的位置为基准点，调整 FY08、FY05 与 FY04

无人机的位置。

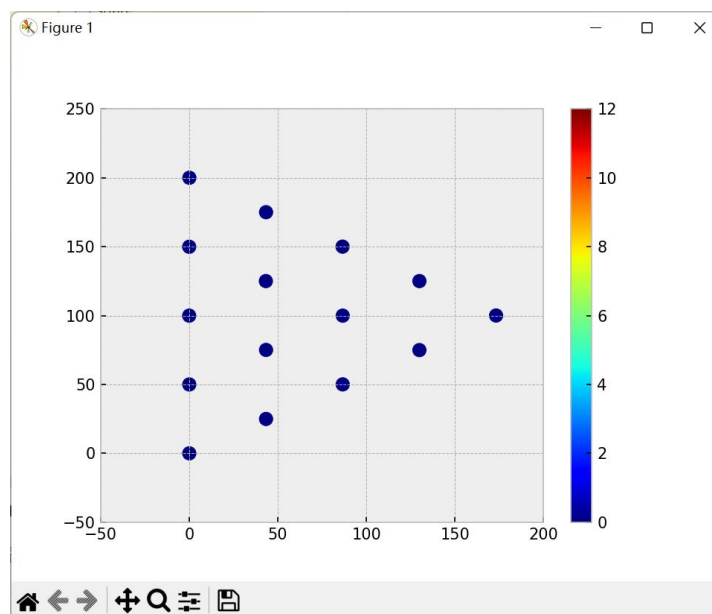
而后又以 FY13、FY08、FY04 无人机的位置为基准点，调整 FY11、FY12、FY07 的无人机位置；以 FY04、FY05、FY06 无人机的位置为基准点，调整 FY01、FY02、FY03 的无人机位置。

最终使得所有无人机的位置都处于一个理想位置之上。

调整前：



调整后：



通过程序运行得到上述两幅图像，可看出无人机位置调整后更为准确（准确度随右侧颜色自下往上而降低）。

六、模型优缺点

6.1 模型的优点

1. 问题一中考虑到了被动机位于三架主动机角平分线上位置不确定的特殊

情况,故提出假设已知主动机发射的次序,使模型更具完备性。

2. 问题二给出了证明:至少需要一架(不包括题目所给的 FY00 和 FY01 两架位置准确且能发射信号的无人机)无人机发射信号即可确定接收信号的无人机的位置。

3. 问题三中建立局部最优模型对无人机位置进行调整,综合模拟结果看,虽然建立的是局部模型,但可以达到全局最优。

4. 问题四建立的模型泛化能力较强,同样适用于其他二维的无人机集群队列进行调整,仅需知道其中两架无人机的精确位置及其他无人机的理想位置,即可对所有有偏差的无人机进行调整,并保证误差小于 1m。

6.2 模型的缺点

1. 利用计算机最小化控制无人机位置调整的目标函数是在初始点进行暴力搜索,即是离散化过程,运算量也与步长设置密切相关。

2. 问题四建立的模型其精确度与运算复杂度呈正相关,即更高的精确度要求更高的运算力,而若是精确度要求到厘米甚至毫米级别时,计算机执行程序的速度也会明显降低,无法实现快速的精准定位。

七、参考文献

[1] 吕佩霞,赵越,李赞,等.主被动协同定位空能资源优化配置方法[J/OL].西安电子科技大学学报:1-9[2024-04-13].<https://doi.org/10.19665/j.issn1001-2400.20240102>.

[2] 刘佳,李宜超.基于无人机纯方位无源定位的研究[J].电子制作,2023,31(21):116-119.DOI:10.16589/j.cnki.cn11-3571/tn.2023.21.024.

[3] 薛振森,郑柏慧,陈礼根,等.基于几何建模与仿真模拟的无人机无源定位模型[J].数学建模及其应用,2023,12(01):79-87.DOI:10.19943/j.2095-3070.jmmia.2023.01.10.

八、附录

附录 1

介绍: question3.py, 针对问题三的局部调整仿真定位

#以 258 为发射点, 点 p 作为接收点的调整策略, 调整策略仅依据三个接收角, 该调整为一轮调整中的第一部分

```
def f258 p(p):
#局部调整一个点(以 258 为发射点调整接收点 3)
#一次调整的过程
#三个接收角度

angle0p2 = calculate_angle(x_lst[p], y_lst[p], x_lst[0], y_lst[0], x_lst[2], y_lst[2])
angle0p5 = calculate_angle(x_lst[p], y_lst[p], x_lst[0], y_lst[0], x_lst[5], y_lst[5])
angle0p8 = calculate_angle(x_lst[p], y_lst[p], x_lst[0], y_lst[0], x_lst[8], y_lst[8])
angle0p2_acc = angle0p2
angle0p5_acc = angle0p5
angle0p8_acc = angle0p8
objective_fp = (max(angle0p2, angle0p5, angle0p8)-70)**2+(mid(angle0p2, angle0p5, angle0p8)50)**2+(min(angle0p2, angle0p5, angle0p8)-10)**2

objective_min_fp = objective_fp
objective_fp_lst=[]
i_min=10
j_min=10
for i in range(0, 21):
    for j in range(0, 21):
        tmp_x=x_lst[p]+0.01*(i-10)
        tmp_y=y_lst[p]+0.01*(i-10)
        angle0p2 = calculate_angle(x_lst[p], y_lst[p], x_lst[0], y_lst[0], x_lst[2], y_lst[2])
        angle0p5 = calculate_angle(x_lst[p], y_lst[p], x_lst[0], y_lst[0], x_lst[5], y_lst[5])
        angle0p8 = calculate_angle(x_lst[p], y_lst[p], x_lst[0], y_lst[0], x_lst[8], y_lst[8])
        objective_tmpp = (max(angle0p2, angle0p5, angle0p8)-70)**2+(mid(angle0p2, angle0p5, angle0p8)-50)**2+(min(angle0p2, angle0p5, angle0p8)-10)**2
        #objective_f3_lst.append((max(angle032, angle035, angle038)-70)**2+(mid(angle032, angle035, angle038)-50)**2+(min(angle032, angle035, angle038)-10)**2)
        if objective_tmpp<objective_min_fp:
            objective_min_fp = objective_tmpp
            j_min = j
            i_min = i
```

```

angle0p2_acc = angle0p2
angle0p5_acc = angle0p5
angle0p8_acc = angle0p8
x_lst[p] += 0.01*(i_min-10)
y_lst[p] += 0.01*(j_min-10)
#print('p: angle0p2_acc, angle0p5_acc, angle0p8_acc, objective_fp, obj
Ective_min_fp)

```

附录 2

介绍: question4.py, 针对问题四调整模型的仿真模拟

```

import numpy as np
import cmath
from matplotlib import pyplot as plt
from matplotlib import colors
from cycler import cycler

# plt_location_advanced 中散点图点的大小
size = 75
# 无人机理想间距
distance_ideal = 50
# 随机数生成种子
seed = 1024
rng = np.random.default_rng(seed)
# 偏差量
offset = 8

# 角度计算函数
def angle(ver, tra1, tra2):
    a = distance(tra1, tra2)
    b = distance(tra1, ver)
    c = distance(tra2, ver)
    d = 2 * b * c
    return np.degrees(np.arccos((b ** 2 + c ** 2 - a ** 2) / d))

def cost_sq(angle1, angle2):
    return np.sum((np.array(angle1) - np.array(angle2)) ** 2)

def distance(p1, p2):
    x = p1[0] - p2[0]
    y = p1[1] - p2[1]

```



```

        return np.sqrt(x ** 2 + y ** 2)

# 极坐标转直角坐标
def polar_to_cartesian(loc): # 极坐标转换为直角坐标
    kid = np.radians(loc[1]) # 角度转化为弧度
    return [loc[0] * np.cos(kid), loc[0] * np.sin(kid)]
    # Input array in radians, output ndarray.

# 直角坐标转极坐标
def cartesian_to_polar(loc):
    if loc[1] < 0:
        loc[1] = loc[1] + 360
    elif loc[1] > 360:
        loc[1] = loc[1] - 360
    cn = complex(loc[0], loc[1])
    return cmath.polar(cn)

# 画出坐标
def plt_location(loc):
    x = []
    y = []
    for i in loc:
        x.append(i[0])
        y.append(i[1])
    plt.scatter(x, y)
    plt.show()

# 显示坐标，并且用颜色代表差异
def plt_location_advanced(loc):
    x = []
    y = []
    t = []
    j = 0
    for i in loc:
        x.append(i[0])
        y.append(i[1])
        t.append(distance(i, location_ideal_cartesian[j]))
        j += 1
    # 美化
    plt.style.use('bmh')
```

```

plt.rcParams['axes.prop_cycle'] = cycler(color='bgrcmyk')
plt.rcParams['image.cmap'] = 'jet'

change_color = colors.Normalize(vmin=0, vmax=12)
plt.scatter(x, y, s=size, c=t, alpha=1, norm=change_color)
plt.xlim((-50, 200))
plt.ylim((-50, 250))
plt.colorbar()
plt.show()

x_distance = np.sqrt(distance_ideal ** 2 - (distance_ideal / 2) **
2)
# 无人机理想直角坐标, 以 FY15 为原点
location_real_cartesian = []
# 无人机理想直角坐标
location_ideal_cartesian = [
    # 第五列
    [x_distance * 4, 100],
    # 第四列
    [x_distance * 3, 125],
    [x_distance * 3, 75],
    # 第三列
    [x_distance * 2, 150],
    [x_distance * 2, 100],
    [x_distance * 2, 50],
    # 第二列
    [x_distance, 175],
    [x_distance, 125],
    [x_distance, 75],
    [x_distance, 25],
    # 第一列
    [0, 200],
    [0, 150],
    [0, 100],
    [0, 50],
    [0, 0],
]

# 给除了 FY15 的飞机添加随机偏移
for i in location_ideal_cartesian:
    if i == [0, 50] or i == [0, 0]:
        x_offset = 0
        y_offset = 0

```

```

        else:
            x_offset = (2 * offset * rng.random() - offset)
            y_offset = (2 * offset * rng.random() - offset)
            location_real_cartesian.append([i[0] + x_offset, i[1] +
y_offset])
        print(location_ideal_cartesian)
        print(location_real_cartesian)

# 用角度确定坐标
def location(fy_id, loc_real, sender1, sender2, sender3, mode):
    if mode == 0:
        angle_real = [angle(loc_real, sender1, sender2),
angle(loc_real, sender2, sender3)]
    elif mode == 1:
        angle_real = loc_real
    # 获取理想位置
    loc_ideal = location_ideal_cartesian[fy_id]
    # 生成初始位置表
    distance_max = 8
    distance_num = 160
    x = np.linspace(-distance_max, distance_max, distance_num,
endpoint=True)
    y = np.linspace(-distance_max, distance_max, distance_num,
endpoint=True)
    # 生成调整后位置表
    x += loc_ideal[0]
    y += loc_ideal[1]
    # 计算理想角度：注意，这个功能也许会在 adjust 中用到
    # angle_ideal = [angle(loc, sender1, sender2), angle(loc,
sender2, sender3)]
    result = 999999
    x_result = 0
    y_result = 0
    # 计算角度差值
    for i in x:
        for j in y:
            loc_calculate = [i, j]
            angle_calculate = [angle(loc_calculate, sender1,
sender2), angle(loc_calculate, sender2, sender3)]
            cost_result = cost_sq(angle_calculate, angle_real)
            if result >= cost_result:
                result = cost_result
                x_result = i

```

```

        y_result = j
    return [x_result, y_result]

def adjust(fy_id, s1, s2, s3):
    loc = location_ideal_cartesian[fy_id]
    sender1_r = location_real_cartesian[s1]
    sender2_r = location_real_cartesian[s2]
    sender3_r = location_real_cartesian[s3]
    sender1 = location_real_cartesian[s1]
    sender2 = location_real_cartesian[s2]
    sender3 = location_real_cartesian[s3]
    angle_ideal = [angle(loc, sender1, sender2), angle(loc, sender2,
sender3)]
    return location(fy_id, angle_ideal, sender1_r, sender2_r,
sender3_r, 1)

def adjust_delta(a1, a2, a3, b1, b2, b3, mode):
    a1 -= 1
    a2 -= 1
    a3 -= 1
    b1 -= 1
    b2 -= 1
    b3 -= 1
    location_real_cartesian[b1] = adjust(b1, a1, a2, a3)
    location_real_cartesian[b2] = adjust(b2, a1, a2, a3)
    location_real_cartesian[b3] = adjust(b3, a1, a2, a3)
    if mode == 0 :
        location_real_cartesian[a3] = adjust(a3, a1, a2, b1)

plt_location_advanced(location_real_cartesian)

adjust_delta(15, 14, 13, 9, 10, 6, 0)
adjust_delta(13, 9, 6, 8, 5, 4, 1)
adjust_delta(13, 8, 4, 12, 7, 11, 1)
adjust_delta(4, 5, 6, 2, 3, 1, 1)

dis = []
j = 0
for i in location_real_cartesian:
    dis.append(distance(i, location_ideal_cartesian[j]))

```

```
j += 1  
print(sum(dis))  
plt_location_advanced(location_real_cartesian)
```