

基于三角定位模型和近似迭代算法的无人机定位与调整

摘要

本文主要解决无人机在编队飞行过程中，为保持集群尽量少向外发射电磁信号，仅选取少量无人机发射信号，其余无人机被动接收信号，如何通过纯方向无源定位来实现无人机定位以及略微偏差的编队队形的调整的问题。

问题一的第一小问中，我们构建了三角定位函数模型，通过定弦定角模型确定出待定位无人机的轨迹，我们通过分类讨论和联立正弦定理方程的方式最终唯一确定待定位无人机的坐标解析式。

问题一的第二小问中，由于发射信号的无人机编号未知引入了新的变量。我们利用第一小问得到的待定位无人机位置信息方程，通过增加发射信号的无人机数增加方程个数，在发射信号的无人机极角度数是 40° 的整数倍的条件下，通过整数规划确定至少新增两架无人机发射信号时，方程组才有唯一解，才能实现无人机的有效定位。

问题一的第三小问中，我们通过近似迭代的方式完成无人机的位置调整。第一步，我们通过近似的方式确认每架无人机的估计坐标，首先随机选取圆周上的三架无人机进行一轮信号发射，并把它们的位置近似到其应在的标准位置。同时我们假设每架待定位无人机都认为自己接受到的信号来源于位置无偏差的发射信号无人机，因此可以得到两个对于该无人机位置的估计值 $(\hat{x}_1, \hat{y}_1), (\hat{x}_2, \hat{y}_2)$ 。对两个估计坐标求平均，将其平均值作为最终的坐标估计 $(\hat{x}, \hat{y}) = (\frac{1}{2} \sum_{i=1}^2 \hat{x}_i, \frac{1}{2} \sum_{i=1}^2 \hat{y}_i)$ 。第二步，我们通过迭代的方式将无人机向其标准坐标逼近，首先根据其编号确定它在编队中的标准坐标 $(R \cos \theta_i, R \sin \theta_i)$ ，再计算出它向标准坐标调整的向量坐标，此外为了防止调整时的每一步过于激进或者过于保守，在向量前乘上学习率 α 以保证收敛。重复以上两步进行近似迭代，并选择合适的学习率，可以在20步迭代左右即可完成准确定位。

问题二中，我们联系“锥形”队形与第三小问中对圆形队形的相似性和不同点，将正六边形等价视为圆形。锥形中部三个相互重叠的六边形使需要作为信号源的无人机架数大大减少，利用锥形中心等边三角形三个顶点处的无人机作为信号源，将他们的坐标近似到标准位置，同样应用三角定位模型和近似迭代调整算法进行模拟。利用重复规则图形共顶点的特性，算法从锥形中心蔓延至锥形的顶点，最终实现整个锥形全部无人机位置的矫正。

本文中我们构建的编队队形调整模型不依赖于特定队形和无人机的数目，具有良好的普适性，可推广到任意规则重复的无人机集群编队调整问题中。同时，模型在不同的初值条件下都有极好的稳定性，在恰当选取迭代调整步长的学习率的情况下，能在一个较少总调整次数后均匀分布到一个圆周上。

关键词: 纯方位无源定位 三角定位 近似迭代

一、问题重述

1.1 问题背景

在无人机集群在编队遂行的过程中，每一架无人机如何调整自身位置以维持整个编队队形保持不变是最主要的问题。若采用传统有源定位方式，即由每一架无人机自身向外主动发射电磁信号从而获取定位，极易受外部环境电磁干扰。而无源定位方式，能在保持固定队形的同时，让尽可能少的无人机发射信号，其余无人机被动接收信号，最大程度避免外界的干扰。接收信号的无人机仅能获得纯方向信息，即该无人机与任意两架发射信号的无人机连线之间的夹角。

1.2 问题重述

- 问题一：9架无人机均匀分布在圆周，1架无人机在圆心，共10架无人机在同一高度构成一个水平圆形编队。第一小问：已知3架发射信号的无人机编号（位置无偏，其中1架在圆心），构建模型以确定其余7架接收信号的无人机（位置略偏）的坐标。第二小问：已知2架发射信号的无人机编号（其中1架在圆心），根据模型，确定至少还需要几架才能准确定位某接收信号的无人机。第三小问：已知10架无人机的初始位置（略偏），每次至多选择4架无人机（其中1架在圆心）发射信号，在尽量少向外发射信号的原则下，给出调整方案，使得在若干次调整后，9架无人机均匀分布在某一圆周上。
- 问题二：与问题一第三小问类似，给出调整方案，使得13架无人机中直线上相邻的两架无人机间距相等，构成锥形编队。

二、问题分析

2.1 问题一的分析

2.1.1 第一小问

要建立接收信号的无人机的定位模型，即通过信号源的位置坐标信息以及接收到的方向角信息确定其极坐标，包括（极径，极角）。已知发射信号的无人机编号，且3架无人机位置均无偏差，则3架无人机的极坐标已知。同时，3架无人机中任意2架与待定位无人机的3个夹角已知，都为定角。而圆周上的无人机与圆心处 $FY00$ 的连线长度为定值（圆的半径），由定弦定角定理，待定位无人机 $FY0X$ 的运动轨迹在一个以 $FY00$ 和圆周上一个确定的无人机 $FY0S_1$ 为弦的圆。同理 $FY0X$ 也同时在 $FY00$ 和圆周上另一无人机 $FY0S_2$ 为弦的圆上运动。通过正弦定理和分类讨论，我们可以求出待定位无人机位置极坐标的解析解，从而唯一确定待定位无人机的坐标位置。

2.1.2 第二小问

比较其与第一小问的不同，已知的信息从三个无人机的位置坐标，变成两架 $FY00$ 和 $FY01$ 。其余新增的用以发射信号的无人机已知极径，极角仅能确定为 40° 的整数倍。因而如果仍旧采用第一小问的方式，在确定圆的轨迹方程时多了一个未知的参数（新增无人机的极角），从而无法唯一确定待定位无人机的坐标。为确定接收信号的无人机的极坐标，我们需要建立与其极径和极

角相关的方程。为了寻找最少需要增加的架数，我们从3架开始，依次往上递增。承袭第一小问的思想，每新增一架便利用其坐标（带未知极角）和FY00和FY01三点进行定位，给出一个带未知参数的待定位无人机极坐标。随着发射信号无人机数目的增多，我们得到的带未知参数的待定位无人机极坐标也越来越多，利用这些坐标构建多个不定方程（方程的解有整数约束），寻找方程个数和未知数之间的关系，从而确定在几个方程的情况下，方程组有唯一解。

2.1.3 第三小问

第三小问是在一二小问的基础上进行条件的进一步弱化，换言之，第一第二小问只是第三小问的特殊情况。在前两小问中，作为信号源发射信号的无人机所在位置不存在偏差，即其极坐标已知。但在第三小问中，信号源无人机的位置也不在队形规定位置，且每一架无人机无法得知自己的初始位置坐标。待定位飞机在每一轮调整中仅能获得的信息，是来源于三架或四架位置存在偏差的无人机的方向角。问题的总目标是使得外面9架无人机均匀分布在某一个圆周上的同时尽可能保证无人机集群尽可能少的向外发射电磁信号，即且在总的调整次数和发射信号无人机架数尽可能少。基于无人机的初始位置“略有偏差”的基本信息，我们可以将初始选定作为信号源的无人机的位置近似成标准点位，每一轮调整中，随机选择出三架无人机作为信号源。待定位无人机都认为他所接收到的方向角信息是来源于位置无偏差的无人机，即其可以通过第一小问中的三角定位法得到一个对自身当前位置的坐标估计 (\hat{x}, \hat{y}) 。以一定的步长向目标位置进行调整。尽管每一次无人机得到关于自己坐标位置的估计都存在偏差，但在迭代若干次之后，随着我们选取的作为信号源的无人机越来越靠近标准位置，即认为这些无人机属于已经被矫正的无人机，其余接收信号的无人机得到的坐标估计与真实坐标间的误差会逐渐减小，最终在有限步迭代后完成对全部无人机的矫正。

2.2 问题二的分析

问题二与问题一第三小问有很大的相似性，在中心的三个三角形周围各有六个等边三角形，构成的正六边形可以等效看成一个圆，而整个锥形是由三个正六边形堆叠而成，因而仅需要中心三架作为信号源，适当改变问题一第三小问的模型，便可确定中心三个六边形内的所有无人机。而顶点的3架无人机在中心六边形的无人机全部矫正之后可以由两点的三角定位直接确定。

三、模型假设

3.1 假设一

待测无人机在知道角度的同时可以提取到角两边的两条射线分别指向哪个无人机。当发射信号的无人机编号已知时，可以确定射线指向的无人机编号；当发射信号的无人机编号未知时，可以确定哪些射线来自同一无人机。该假设对于后续定义顺逆角解决对称造成的必然多解问题有重要意义。

3.2 假设二

无人机调整过程中，在估算自身坐标时，计算出从自身坐标到目标坐标需要移动的方向和需

要移动的距离。该假设对于明确无人机的调整方式不可或缺。

3.3 假设三

四、符号说明

符号	说明	单位
FY0X	无人机编号	
R,r	极径	m
$\theta_0, \theta, \hat{\theta}$	极角	°
$\alpha_1, \alpha_2, \alpha_3$	发射信号无人机相对待定无人机方位的夹角	°
i_1, i_2	对应 α_1, α_2 的顺逆时针	
k_1, k_2	无人机编号对应极角与40°的整数倍	
$(x_i, y_i), (\hat{x}_i, \hat{y}_i), (x, y)$	模拟坐标	m

五、模型的建立与求解

5.1 问题一第一小问

5.1.1 模型的建立

记待定位无人机编号为FY0X，我们不妨令FY01为圆周上一架确定作为发射信号的无人机，因为在此时选择时FY01与圆周上其余八架无人机的地位是平等的，设圆周上另一架信号源无人机为FY0S。设编队队形的圆形的半径为R。

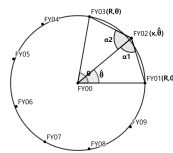
考虑到队形为圆形，用极坐标系更易求解。因此以FY00为极点，FY00指向FY01为极轴的正方向。已知三架信号源无人机的极坐标，分别为

$$FY00 : O = (0, 0) \quad FY01 : I = (R, 0) \quad FY0S : S = (R, \theta)$$

设待定位无人机的极坐标为 $(x, \hat{\theta})$ 在利用极坐标进行求解时，我们利用正弦定理来构建圆的轨迹方程，但由于正弦定理以及三角函数本身在角度不同时所对应符号不同，我们需要进行分类讨论。

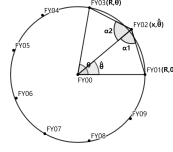
1. 情况一

$$\begin{cases} \frac{\sin \alpha_1}{R} = \frac{\sin(\alpha_1 + \hat{\theta})}{x} \\ \frac{\sin \alpha_2}{R} = \frac{\sin(\theta - \hat{\theta} + \alpha_2)}{x} \end{cases}$$



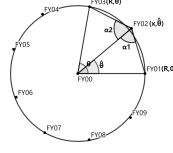
2. 情况二

$$\begin{cases} \frac{\sin \alpha_1}{R} = \frac{\sin(\alpha_1 - \hat{\theta})}{x} \\ \frac{\sin \alpha_2}{R} = \frac{\sin(\hat{\theta} - \theta + \alpha_2)}{x} \end{cases}$$



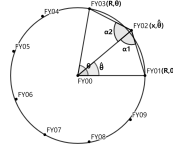
3. 情况三

$$\begin{cases} \frac{\sin \alpha_1}{R} = \frac{\sin(\alpha_1 + \hat{\theta})}{x} \\ \frac{\sin \alpha_2}{R} = \frac{\sin(\hat{\theta} - \theta + \alpha_2)}{x} \end{cases}$$

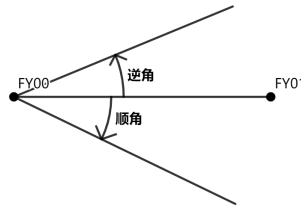


4. 情况四

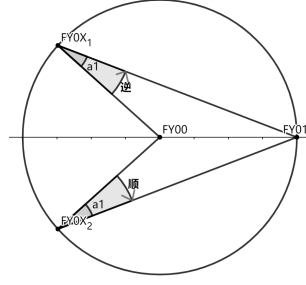
$$\begin{cases} \frac{\sin \alpha_1}{R} = \frac{\sin(\alpha_1 - \hat{\theta})}{x} \\ \frac{\sin \alpha_2}{R} = \frac{\sin(\theta - \hat{\theta} + \alpha_2)}{x} \end{cases}$$



我们对分类的标准重新做了如下规定，使得在此标准下，四种情况的方程式可以归结为一个统一的表达式。基于假设一，待测无人机可以准确得知其接受到的每一个方位角信号是由两架确定编号的无人机发出的，即在知道角度的同时也提取到了角两边的两条射线分别指向哪个编号的无人机。又因为FY00始终确定作为信号源，因此以待测无人机FY0X为起点，做穿过FY00的射线。以这条射线为基准线，三个方位角中的 α_1 和 α_2 看成是这条射线绕FY0X旋转形成的，若顺时针旋转形成则定义为“顺角”，反之，若逆时针旋转则为“逆角”。在此视角下重新观察以上四种情况，我们不难发现，其唯一差别在于方向角 α_1 和 α_2 的顺逆之别。



又由于 α_1 生成与两架我们确定编号和位置的无人机FY00和FY01，以这两架无人机的连线为坐标轴，会将圆分割成“上半圆”和“下半圆”，通过 α_1 的顺逆，无人机可以很便捷地判断出自己的在以FY00和FY01为参考系下的相对位置



进而，我们对四种情况的四个方程式进行统一的归纳：

$$\begin{cases} \frac{\sin \alpha_1}{R} = \frac{i_1 \sin(\hat{\theta} + i_1 \alpha_1)}{x} \\ \frac{\sin \alpha_2}{R} = \frac{i_2 \sin(\hat{\theta} - \theta + i_2 \alpha_2)}{x} \end{cases}$$

其中

$$i_1 = \begin{cases} +1 & \text{当}\alpha_1\text{为逆角} \\ -1 & \text{当}\alpha_1\text{为顺角} \end{cases} \quad i_2 = \begin{cases} +1 & \text{当}\alpha_2\text{为逆角} \\ -1 & \text{当}\alpha_2\text{为顺角} \end{cases}$$

我们做此规定是为了解决圆形图案中存在的“对称问题”。在仅知三个方向角和对应信号源编号时，由于图形的对称性，方程总是存在两解的情况，因而我们对方向角信息进行深度挖掘和利用，从而在不增加额外假设和信息的情况下，使得待测无人机对于自身与三架发射信号的无人机在队形中的相对位置有一个初步的认知，这个认知对于判断如何舍去方程的解意义重大，从而可以进一步实现唯一、有效、精准的定位。

5.1.2 模型的求解

模型二

求解方程

$$\begin{cases} \frac{\sin \alpha_1}{R} = \frac{i_1 \sin(\hat{\theta} + i_1 \alpha_1)}{x} \\ \frac{\sin \alpha_2}{R} = \frac{i_2 \sin(\hat{\theta} - \theta + i_2 \alpha_2)}{x} \end{cases}$$

并化简得

$$\tan \hat{\theta} = \frac{i_2 \sin \alpha_1 \sin(i_2 \alpha_2 - \theta) - i_1 \sin \alpha_2 \sin(i_1 \alpha_1)}{i_1 \sin \alpha_2 \cos(i_1 \alpha_1) - i_2 \sin \alpha_1 \cos(i_2 \alpha_2 - \theta)}$$

进一步化简合并得

$$\tan \hat{\theta} = \frac{i_2 \sin \alpha_1 \sin(i_2 \alpha_2 - \theta) - \sin \alpha_1 \sin \alpha_2}{i_1 \sin \alpha_2 \cos \alpha_1 - i_2 \sin \alpha_1 \cos(i_2 \alpha_2 - \theta)}$$

每架无人机在根据方位角发掘出 α_1 和 α_2 的顺逆之后，便可以唯一确定 i_1 和 i_2 的值，由此代入上面的解析解可以唯一确定 $\tan \hat{\theta}$ 的值。再根据 α_1 的顺逆，判断出待定位无人机在以FY00和FY01为极轴的坐标系中位于上半圆还是下半圆，从而极角 $\hat{\theta}$ 可以根据 $\tan \hat{\theta}$ 的值唯一确定，继而极径 x 也可唯一确定。

5.2 问题一第二小问

为解决至少需要在圆周上增加几架无人机发射信号，才能实现对其余接收信号的无人机的有效定位问题，我们采取依次递增的方式。由于已经确定FY00和FY01作为信号源，每次在圆周上增加一架，试证明在充分挖掘利用所有已知信息的情况下，能否给出其余无人机的唯一、确定的定位。

1. 新增一架

圆周上新增1架编号为FY0S₁，但其编号对于接收信号的无人机是未知的，即在定位过程中无法使用其坐标信息。我们可将未知编号无人机的坐标表示为 $(R \cos(k_1 \theta_0), R \sin(k_1 \theta_0))$ ，其中 k_1 为整数， $\theta_0 = 40^\circ$ ，即虽然我们不知道他的编号，但其仍然是圆周上一架位置无偏差的无人机。但由于其编号未知，第一小问中所得公式中的 θ 成为未知数，方程存在多组关于位置参数 θ 的解。因此可以判定新增一架未知编号无人机无法实现有效定位。

2. 新增两架

假设新增的两架发射信号的无人机为FY0X ($R, k_1 \times 40^\circ$)、FY0Y ($R, k_2 \times 40^\circ$)，接收信号的无人机从FY00、FY01、FY0X接收到的方向信息为 α_1 、 α_2 、 α_3 ，从FY00、FY01、FY0Y接收到的方向信息为 β_1 、 β_2 、 β_3 。在上一问求得的公式里，通过 α_1 的顺/逆可以判断出 i_1 等于1还是-1以及 $\hat{\theta}$ 是小于还是大于 180° 。将 α_1 、 α_2 、 α_3 代入公式1可得：

$$\tan \hat{\theta} = \frac{i_2 \sin \alpha_1 \sin(i_2 \alpha_2 - k_1 \times 40^\circ) - \sin \alpha_1 \sin \alpha_2}{i_1 \sin \alpha_2 \cos \alpha_1 - i_2 \sin \alpha_1 \cos(i_2 \alpha_2 - k_1 \times 40^\circ)}$$

将 β_1 、 β_2 、 β_3 代入公式1可得：

$$\tan \hat{\theta} = \frac{t_2 \sin \beta_1 \sin(t_2 \beta_2 - k_2 \times 40^\circ) - \sin \beta_1 \sin \beta_2}{i_1 \sin \beta_2 \cos \beta_1 - t_2 \sin \alpha_1 \cos(t_2 \beta_2 - k_2 \times 40^\circ)}$$

其中 $i_1 = \pm 1$ ， $i_2 = \pm 1$ ， $t_2 = \pm 1$ ， k_1 、 k_2 属于 $\{1, 2, 3, 4, 5, 6, 7, 8\}$ 且 k_1 不等于 k_2 。联立这两个方程式，在其他都是变量都是常数的条件下，不难发现该整数规划问题只可能有唯一解。

综上所述，还需要2架无人机才能实现接收信号无人机的有效定位。

5.3 问题一第三小问

5.3.1 模型的建立

我们将每一轮调整方案分为1.选择发射信号无人机和2.调整接收信号无人机两步。

• 选择发射信号无人机

在每一轮可以选择不同的三架无人机作为信号源，由于发射信号的无人机是其余接收信号无人机定位自身位置坐标的唯一基准，因此信号源无人机选取的优劣与否直接决定了每一步调整的误差大小，以及迭代算法最终能否收敛。对于圆周上无人机数量的选择，通过第一小问我们可以知道，在已知两架在标准位置上无人机编号的前提下可以得出平面上任意一点的坐标位置。由于我们采取以标准位置近似实际位置的方法来确定发射信号的无人机位置，因此通过此近似方法结合第一小问的方法求得的待定位无人机坐标存在一定程度的

误差。而当使用三架圆周上的无人机时，可以任选两次其中的两架无人机，用近似的方法求出两个待定位无人机的近似位置坐标的平均值以减小误差。相比而言，圆周上选择两架无人机的优势是可以减少电磁辐射，但是劣势是精度较低，而三架无人机则与之相反，因此可以选择在迭代开始阶段选择三架无人机，而在迭代后期误差较小时可以选择两架无人机。基于无源定位的基本假设，我们规定作为信号源发射的无人机在本轮无法接收信号。因此，在无法获得无人机集群的全局统计信息时，对于每一架无人机来说，位置均存在偏差，因此随机化的选择是可行的、公平的。

我们在每轮均随机从圆周上9架无人机中选取其中3架作为作为信号源。

```
1 indexes = sort(randperm(9));
```

同时，随着无人机群体在协同调整过程中逐渐趋于圆形，信号源无人机的坐标不断逼近无偏差的标准位置，最终所有的无人机都会被矫正，算法最终可以保证收敛。

• 调整接收信号无人机

在选定了圆周上作为信号源的4架无人机的编号 $FYOS_1$ ， $FYOS_2$ ， $FYOS_3$ 后，我们将这几架略有偏差的无人机近似到其标准位置。同时，我们假设每一架待定位无人机都认为自己接收到的信号来源于位置无偏差的发射信号无人机，此时情况与第一小问一致。我们从三架中每次选其中两架，结合 $FY00$ ，即已知三个方位角信息和三个信号源坐标，代入第一小问的定位模型函数中，我们可以得到对该架无人机位置的两个估计值

$$(\hat{x}_1, \hat{y}_1), \quad (\hat{x}_2, \hat{y}_2)$$

对两个坐标估计求平均，将平均值作为最终的坐标估计

$$(\hat{x}, \hat{y}) = \left(\frac{1}{2} \sum_{i=1}^2 \hat{x}_i, \frac{1}{2} \sum_{i=1}^2 \hat{y}_i \right)$$

三角定位模型函数

```
1 function coordinates = get_coordinates(index,alpha,clockwise,data_std)
2 % 通过输入角度求出坐标
3 i = -clockwise;
4 coordinates = [];
5 % 第k个飞机的两个，以及对应的角度kalpha
6 for k = 1:9
7     if k == index(1) || k == index(2)
8         coordinates = [coordinates;data_std(k,:)];
9     else
10         theta = (index(2)-index(1))*40;
11         tan_theta_hat = ((sind(alpha(k,1))*i(k,2)*sind(i(k,2)*alpha(k,2)-theta)-sind(alpha(k,2))*sind(alpha(k,1)))/ ...
12             (sind(alpha(k,2))*i(k,1)*cosd(alpha(k,1))-sind(alpha(k,1))*i(k,2)*cosd(i(k,2)*alpha(k,2)-theta)));
13         if i(k,1) >= 0
14             if tan_theta_hat >= 0
15                 theta_hat = atan(tan_theta_hat)/pi*180;
```



```

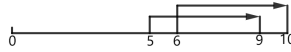
16         else
17             theta_hat = atan(tan_theta_hat)/pi*180+180;
18         end
19     else
20         if tan_theta_hat>=0
21             theta_hat = atan(tan_theta_hat)/pi*180+180;
22         else
23             theta_hat = 360+atan(tan_theta_hat)/pi*180;
24         end
25     end
26     theta_hat = theta_hat+(index(1)-1)*40;
27     r = 100*i(k,1)*sind(theta_hat+i(k,1)*alpha(k,1))/sind(alpha(k,1));
28     coordinates = [coordinates;r*cosd(theta_hat) r*sind(theta_hat)];
29 end
30 end
31 end

```

该无人机的调整的目标坐标为其编号所对应的编队中的标准坐标($R \cos \theta_i, R \sin \theta_i$), 其中 θ_i 为待定位无人机的编号在编队中所对应的标准位置。例如,

FY02 的标准位置为 ($100 \cos 40^\circ, 100 \sin 40^\circ$)

基于假设二, 无人机可以在已知自身坐标和目标坐标点情况下直接移动到目标位置。但由于无人机计算得到的对自身位置坐标的估计存在误差, 因此在测算调整移动的方向和距离时均存在误差。举一维的情况说明, 无人机的实际坐标位置为5, 其调整的目标位置为10, 但其对自身坐标位置的估计为6, 通过计算只要移动4即可达到, 但实际只移动到了9, 依旧无法到达。



为了在每一步调整时既不过于激进也不过于保守, 在临近标准位置时反复越过导致最终算法无法收敛, 我们在每一步调整的步长前附加了一个大小为 α 的学习率。因此, 在每一轮选定信号源无人机后, 对于剩余的每一架接收信号的无人机的调整方案为:

1. 将方向角信息代入三角定位模型, 得到对自身坐标的估计(\hat{x}, \hat{y})
2. 接收信号无人机向自身编号在编队中的标准位置移动, 移动方向为向量 \vec{s} 的方向

$$\vec{s} = (R \cos \theta_i, R \sin \theta_i) - (\hat{x}, \hat{y})$$

3. 移动的距离为(其中 $\alpha \in (0, 1)$)

$$\alpha \|s\|_2 = \alpha \sqrt{(R \cos \theta_i - \hat{x})^2 + (R \sin \theta_i - \hat{y})^2}$$

接收信号无人机坐标调整函数

```

1 function data_adjusted = adjust_of_drone(data,data_std)
2 % 每次调整，通过三四架飞机来确定当前飞机的位置坐标/
3 % 再用估计坐标调整到标准位置上
4 data_adjusted = data;
5 learning_rate = 0.1;
6 % indexes = choice_of_drone(data,data_std);
7 indexes = sort(randperm(9));
8 for i = 1:500
9     [alpha_case12,clockwise12] = get_alpha(data_adjusted,[indexes(1),indexes(2)]);
10    [alpha_case13,clockwise13] = get_alpha(data_adjusted,[indexes(1),indexes(3)]);
11    coordinates12 = get_coordinates([indexes(1),indexes(2)],alpha_case12,clockwise12,
        data_std);
12    coordinates13 = get_coordinates([indexes(1),indexes(3)],alpha_case13,clockwise13,
        data_std);
13    coordinates = (coordinates13+coordinates12)/2;
14    data_adjusted = data_adjusted-learning_rate*(coordinates-data_std);
15    %     indexes = choice_of_drone(data_adjusted,data_std);
16    indexes = sort(randperm(9));
17 end
18 end

```

5.3.2 模型的求解

我们利用题目所给的初始位置坐标作为初值，代入模型。在不同的学习率与迭代步长下进行无人机编队队形调整的数值实验。我们将调整后得到的所有无人机的位置坐标拟合成为一个圆形。通过计算残差去观察和比较调整后的队形，即每架无人机是否均匀分布在一个圆周上。

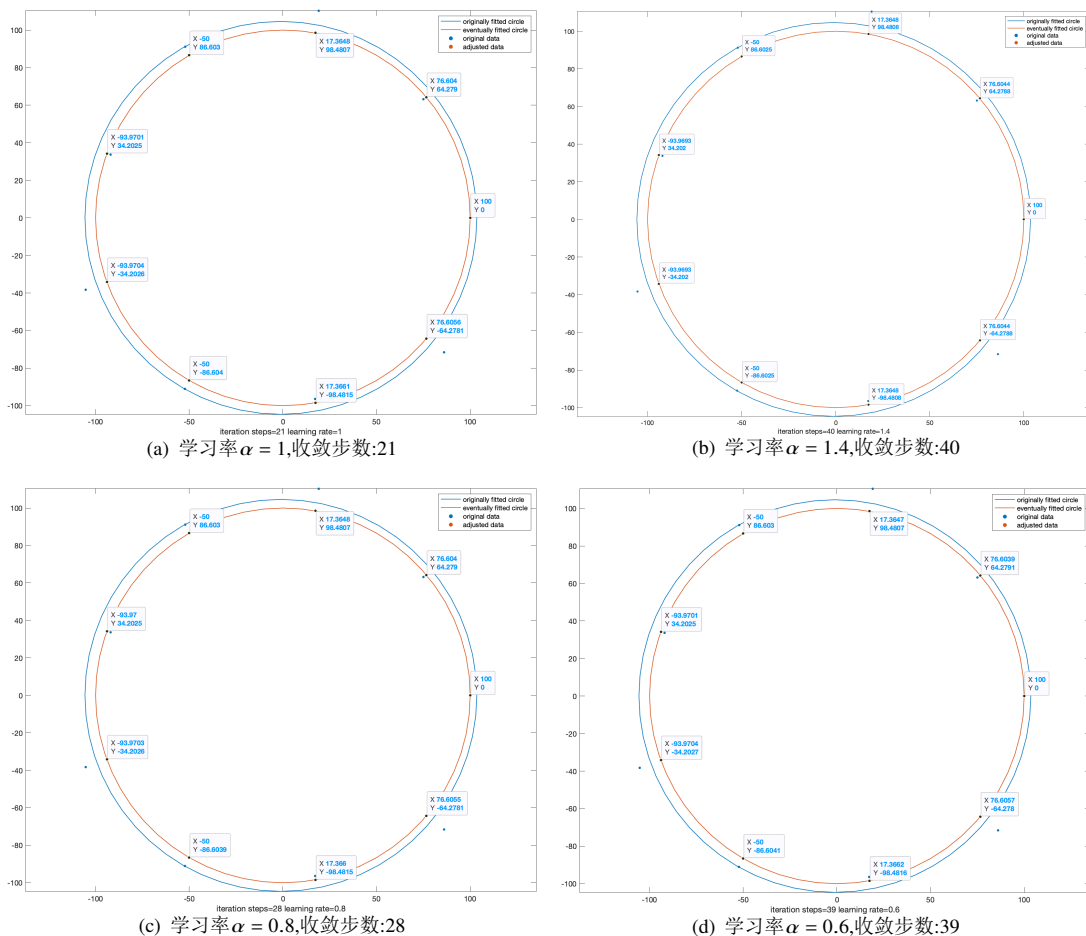
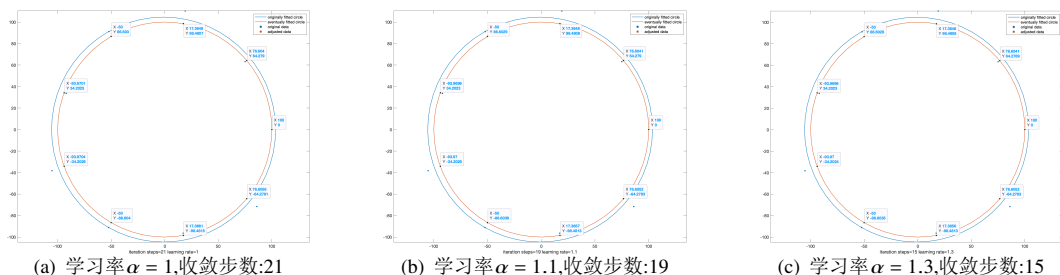


图 1: 不同学习率下收敛步数对比

观察不同学习率下收敛步数，我们不难发现，当 α 取比较小的值时，即对每一次的调整步长都相对保守的情况下，需要40次左右到调整，才能最终收敛。而当 α 取比较大的值时，即对每一次的调整步长都相对激进的情况下，也需要较多的迭代步数（调整次数）才能最终收敛。学习率 $\alpha = 1.3$ ，是针对当前数据集的一个最优取值，即使得总调整次数最少的取值。



比较在同一学习率($\alpha = 0.6$)下不同迭代步数的残差，我们不难发现随着迭代步数的增加，在40步附近，算法收敛，即通过我们的调整方案，能够使得原本位置稍有偏差的9架无人机，经过有限步的调整，最终均匀分布在一个圆周上。

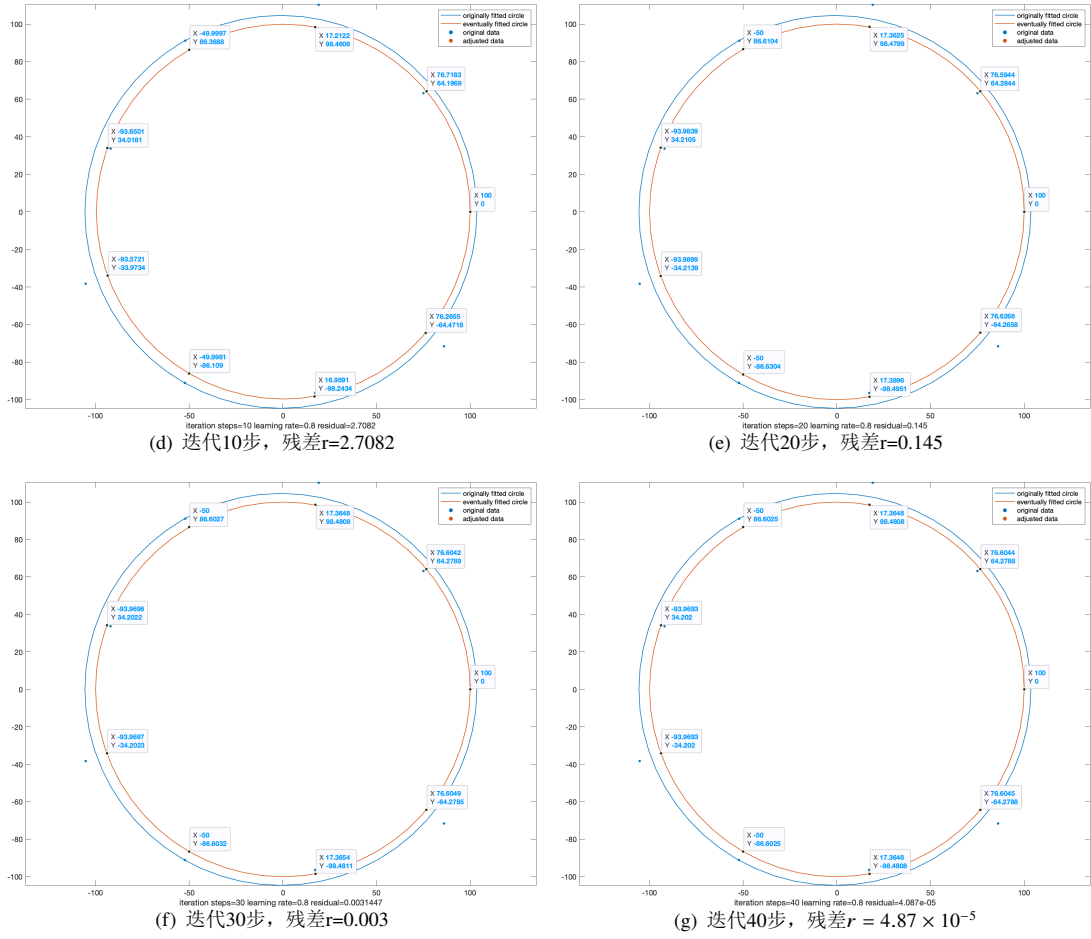
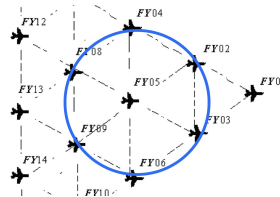


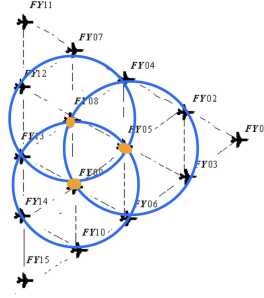
图 2: 同一学习率($\alpha = 0.6$)下不同迭代步数的残差

5.4 问题二模型的建立与求解

由于我们在问题一第一小问中建立的模型并不依赖于整体的圆形队形以及无人机的数量, 因而自然可以将其模型移植到“锥形”编队中。仔细观察我们不难发现, 锥形中有三个互相重叠的正六边形, 每个正六边形由六个正三角形构成。我们可以等价的将六边形的六个顶点放置在一个圆上, 那么同理, 我们可以取圆心和圆周上两架无人机对另外4架无人机实现定位和调整。只不过此时在圆上六架飞机间的均匀间隔变为 60° , 圆的半径变为50



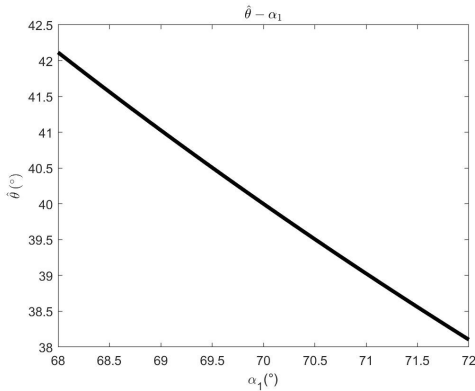
同时, 我们只要取三个互相重叠六边形的三个中心点作为信号源, 便可以全部确定三个六边形的所有顶点上共9架无人机。



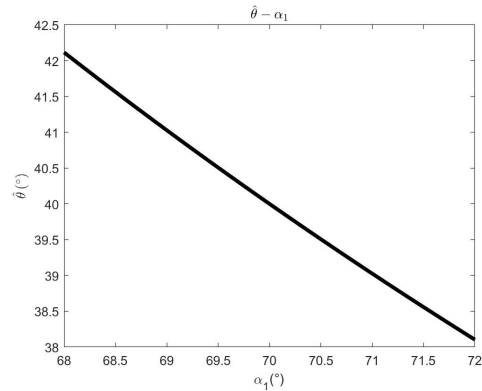
最后“锥形”三个顶点处的无人机，可以利用等边三角形的特殊性，仅需要另其紧邻的三角形的三个顶点作为信号源发射信号，即可实现调整。

六、模型的分析与检验

在问题一（1）中，我们通过严格的数学推导求出了被动接收信号无人机的极坐标公式。因而若能保证无人机接收到的方向信息 α_1 、 α_2 和 α_3 准确无误，且不考虑计算精度的限制，则由该公式得到的被动接收信号无人机的位置信息也是准确无误的。但在实际应用中，由于传感器测向精度有限等因素， α_1 、 α_2 和 α_3 必然存在误差。可以通过改进传感器、优化发射接收方案等手段提高精度，但这不在我们的讨论范围内，故不作展开。需要我们关心的是—— α_1 、 α_2 和 α_3 的误差将会如何影响得到的被动接收信号无人机的位置信息的准确性？查阅资料可知，传统雷达测向精度为 2° – 4° 。以图例所示情况为4种讨论情况的图示1，进行误差估计与灵敏度分析。假设 $R = 100m$ ； $\theta=80^\circ$ ； $\theta < 180^\circ$ ； α_1 、 α_2 准确值为： $\alpha_1=70^\circ$ ， $\alpha_2=70^\circ$ 。容易知道，此时准确的被动接收信号无人机的极坐标 $(x, \hat{\theta})$ 为 $(100m, 40^\circ)$ 。固定 $\alpha_2=70^\circ$ ，让 α_1 在 68° – 72° 变化，观察求出的 x 与 $\hat{\theta}$ 的相应变化情况。



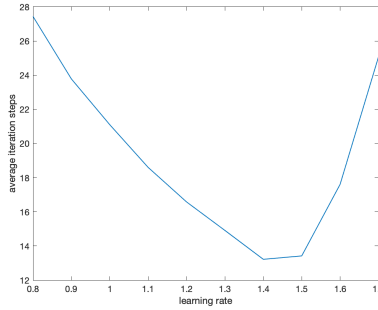
(a) 图示1



(b) 图示2

可以看到，当 α_1 在 68° – 72° 变化时， $\hat{\theta}$ 在 38° – 42.5° 变化， x 在 $98.5m$ – $101.5m$ 变化，与 α_1 大致都呈负线性相关。 α_1 的测向不准引起的 $\hat{\theta}$ 偏离准确值约在 $(-2^\circ, +2.5^\circ)$ 内，引起的 x 偏离准确值约在 $(-1.5m, +1.5m)$ 内。固定 α_1 ，让 α_2 变化时情况大体类似，在此不作赘述。当然， α_1 、 α_2 的误差累积可能会造成 $\hat{\theta}$ 和 x 的更大误差。应当考虑无人机的实际用途（如用于军事领域、无人机飞行表演等）来决定该误差是否在可接受范围内。

在问题一（3）中，生成随机的初始位置数据集，其中角度均值是40度的倍数，方差为1。半径均值为100，方差为4。进行100次实验数值实验，得到不同学习率下收敛所需迭代步数的均值，结果如下。



对于满足该特定分布的100个随机初始位置数据集，学习率 $\alpha \in (1.4, 1.5)$ 之间时，可以获得最快的迭代下降步数。

七、模型的评价

7.1 模型的优点

1. 问题一第一小问中建立的三角定位模型，既针对编队这种队形采用极坐标来简化运算，又在求解待定位无人机的轨迹方程时转化为三角形中的正弦定理，通过普适的边角关系限制动点的运动，充分发挥了极角的作用。
2. 为了解决圆形中存在的“对称问题”，即相同的三架无人机可以向左右两架关于某条轴对称的无人机发射完全相同的三个方向角信息，即仅根据一组定弦定角确定出的轨迹方程分为上半圆弧与下半圆弧两个部分。因此，我们设计出通过判定 α_1 为“顺角”和“逆角”来大致判断自身与发射信号的三架无人机相对位置关系，若以其中某两架为坐标轴建立坐标系，则可以进一步得到自己所处“上半圆”或“下半圆”的信息。这个信息是对方向角的深度挖掘，对于唯一确定方程的解意义重大。
3. 在调整接收信号无人机时，我们做出了近似假设和学习率限制。近似假设在初始位置“略有偏差”的背景下能够大大减少计算复杂度，且在信号源飞机接近矫正时往后每一步迭代的误差都会缩小。学习率对调整步长的限制保证了无人机在调整时不会过于激进，防止在临近标准位置附近反复移动错过，使得算法无法收敛。
4. 问题一第三小问所设计出的调整方案具有极佳的普适性质，其并不依赖于编队中无人机的数量以及编队的队形。可以适用于所有重复的规则队形。

7.2 模型的缺点

1. 在采用极坐标和正弦定理的同时，不可避免需要对不同的夹角情况进行分类讨论，尽管通过规定，可以将四种情况归结为统一的表达式，但在判定系数符号的过程依旧稍显复杂，

在模型改进中，我们通过用平面直角坐标加余弦定理来规避符号判定与针对不同具体情况的分类讨论。

2. 在“稍有偏差”的背景下，采用全随机的选取方法，所带来的误差可能会累加。若在真正的调整轮次前增加了一个预发射轮次以通过改良的贪心算法确定最优的信号源编号，随之带来的是信号发射次数多增加，我们希望通过每一步都贪心选择来减少总体的迭代次数。

7.3 模型的改进

模型一：平面直角坐标模型

以FY0X为原点建立平面直角坐标系，已知三架信号源无人机的编号即已知三架无人机的平面直角坐标，分别为

$$FY00 : O = (0, 0) \quad FY01 : I = (R, 0) \quad FY0S : S = (R \cos \theta, R \sin \theta)$$

同时，由假设一，我们可以作出如下规定：

FY01和FY00与FY0X连线之间的夹角为 α_1

FY0S和FY00与FY0X连线之间的夹角为 α_2

FY01和FY0S与FY0X连线之间的夹角为 α_3

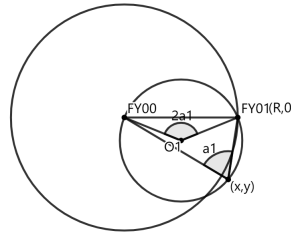
定理：定弦定角定理。已知三角形中一条边的长度为定值（定弦），其所对的角为一个定角（圆周角），则这条边所对的顶点的轨迹方程是圆。

由于FY00和FY01构成的线段长度为圆的半径R，二者与FY0X的夹角为 α_1 ，因此，FY0X的轨迹在一个圆上，设这个圆的圆心 O_1 为 (x_1, y_1) ，半径为 r_1 ，圆的方程为：

$$(x - x_1)^2 + (y - y_1)^2 = r_1^2$$

此外， $\angle OO_1I$ 为弦OI的圆心角，其同弧所对圆周角为 α_1 ，因此，在 $\triangle OIO_1$ 中，利用余弦定理

$$R^2 = r_1^2 + r_1^2 - 2r_1^2 \cos 2\alpha_1$$



同时，利用点O和点I都在圆上，我们可以列出三个方程，从而解得 (x_1, y_1) 和 r_1

$$\begin{cases} (0 - x_1)^2 + (0 - y_1)^2 = r_1^2 \\ (R - x_1)^2 + (0 - y_1)^2 = r_1^2 \\ R^2 = r_1^2 + r_1^2 - 2r_1^2 \cos 2\alpha_1 \end{cases}$$

同理，对于FY0S、FY00和FY0X，也有一组定弦定角(OS; α_2)，可得(x_2, y_2)和 r_2

$$\begin{cases} (0 - x_2)^2 + (0 - y_2)^2 = r_2^2 \\ (R \cos \theta - x_2)^2 + (R \sin \theta - y_2)^2 = r_2^2 \\ R^2 = r_2^2 + r_2^2 - 2r_2^2 \cos 2\alpha_2 \end{cases}$$

对于FY0S、FY01和FY0X，有一组定弦定角(IS; α_3)，可得(x_3, y_3)和 r_3

$$\begin{cases} (R - x_3)^2 + (0 - y_3)^2 = r_3^2 \\ (R \cos \theta - x_3)^2 + (R \sin \theta - y_3)^2 = r_3^2 \\ 2R^2(1 - \cos \theta) = r_3^2 + r_3^2 - 2r_3^2 \cos 2\alpha_3 \end{cases}$$

随后我们联立三个圆的方程，可以得到一组唯一的解，即可以唯一确定出一个交点(x, y)，这个交点就是FY0X的坐标。

$$\begin{cases} (x - x_1)^2 + (y - y_1)^2 = r_1^2 \\ (x - x_2)^2 + (y - y_2)^2 = r_2^2 \\ (x - x_3)^2 + (y - y_3)^2 = r_3^2 \end{cases}$$

求解三组圆心和半径

$$\begin{cases} x_1 = \frac{R}{2} \\ y_1^2 = \frac{R^2}{2(1 - \cos(2\alpha_1))} - \frac{R^2}{4} \\ r_1^2 = \frac{R^2}{2(1 - \cos(2\alpha_1))} \end{cases}$$

$$\begin{cases} x_2 = \frac{R}{2 \cos \theta} - \tan \theta y_2 \\ y_2 = \frac{R \sin \theta \pm \sqrt{4r_2^2 - R^2 \cos \theta}}{2} \\ r_2^2 = \frac{R^2}{2(1 - \cos(2\alpha_2))} \end{cases}$$

$$\begin{cases} x_3 = \frac{R \sin^2 \theta \pm \sin \theta \sqrt{R^2 \sin^2 \theta - 2(R^2 - r_3^2)(1 - \cos \theta)}}{2(1 - \cos \theta)} \\ y_3 = \frac{1 - \cos \theta}{\sin \theta} x_3 \\ r_3^2 = \frac{R^2(1 - \cos \theta)}{1 - \cos(2\alpha_3)} \end{cases}$$

由于圆的对称性，三个圆心坐标不可避免的出现了解的情况。但应用模型一中，我们对 α_1 顺逆的规定，待定位无人机得以进而判断出自己所处上半圆或下半圆，从而可以唯一确定圆心坐标，实现对于对称解的有效舍去。求解联立的三个圆的方程，化简成为矩阵形式

$$\begin{bmatrix} r_1^2 - (x_1^2 + y_1^2) \\ r_2^2 - (x_2^2 + y_2^2) \\ r_3^2 - (x_3^2 + y_3^2) \end{bmatrix} = \begin{bmatrix} -2x_1 & -2y_1 & 1 \\ -2x_2 & -2y_2 & 1 \\ -2x_3 & -2y_3 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ x^2 + y^2 \end{bmatrix}$$

由于(x_i, y_i)仅与 α_i 和 θ 有关，在四个参数已知的情况下为常数，因此可以继续化简成

$$B = AX$$

继而转化为求解最小二乘法的数值解

$$\min_x \|B - AX\|_2^2$$

总的来说，定位模型可以归结为输入三个方位角信息和三个信号源的坐标 $(\alpha_1, \alpha_2, \alpha_3, \theta)$ ，输出待定位无人机坐标 (x, y) 的函数。

调整方案的改进

优化选择发射信号无人机

在每一轮可以选择不同的三架无人机作为信号源，由于发射信号的无人机是其余接收信号无人机定位自身位置坐标的唯一基准，因此信号源无人机选取的优劣与否直接决定了每一步调整的误差大小，以及迭代算法最终能否收敛。基于无源定位的基本假设，我们规定作为信号源发射的无人机在本轮无法接收信号。同时，作为方案的设计者，为了选取最佳的信号源无人机，我们需要在每一轮真正调整前，进行一次预发射，即在圆周上随机选择两架作为信号源，与圆心处无人机共同向其余无人机发射信号，其余无人机始终认为信号源无人机处于无偏差标准位置。由第一小问的结论，我们可以得到除信号源以外7架无人机位置坐标点估计值

$$(\hat{x}_i, \hat{y}_i) \quad i = 1, 2, \dots$$

同时，我们将两架信号源无人机的位置近似为其编号在编队中对应的标准位置，这样我们就能通过外部9架无人机的估计坐标，拟合得到一个圆形，其半径和圆心分别为

$$(x_0, y_0), \quad r_0$$

通过计算比较每架无人机到当前圆心的距离与拟合得到的圆形半径之间的差距，我们可以选出其中差距最小的三架无人机，作为真正调整轮次发射信号的无人机。

$$\min_i \left| r_0 - \sqrt{(\hat{x}_i - x_0)^2 + (\hat{y}_i - y_0)^2} \right|$$

这样选择能始终保证作为信号源的无人机与标准圆形队形的偏差最小，即保证我们在估计每一架接收信号无人机坐标时做出的假设不会带来较大的误差。同时，随着无人机群体在协同调整过程中逐渐趋于圆形，信号源无人机的坐标不断逼近无偏差的标准位置，最终所有的无人机都会被矫正，算法最终可以保证收敛。

参考文献

[1]张笑.无源二维交叉定位分析[J].电子科技,2013,26(02):54-56+59.DOI:10.16180/j.cnki.issn1007-7820.2013.02.007.

附录

附录1

程序源代码: *approximate_iteration.m*

附录2

该程序用MATLAB编写，其中主要函数包括：问题一第一小问中的三角定位函数，问题一第三小问中对接收信号无人机的坐标估计函数以及近似迭代调整函数。

```
1 clear;
2 clc;
3
4 data = [100,0;
5         74.9623,63.1241;
6         19.0442,110.3690;
7         -52.1027,91.1609;
8         -92.0077,33.7429;
9         -105.2723,-38.2328;
10        -52.3889,-90.9967;
11        17.3038,-96.4602;
12        86.1478,-71.5721]; % 没有选取(,)点00
13
14 % 随机生成测试数据
15 data_std = zeros(9,2);
16 for i = 0:8
17     ang = 40*i;
18     data_std(i+1,:) = [100*cosd(ang) 100*sind(ang)];
19 end
20
21 dddata = [];
22 for learning_rate = 0.9:0.1:1.6
23     sums = 0;
24     for j = 1:100
25         % 随机生成初始数据点
26         data_test = zeros(9,2);
27         data_test(1,:) = [100 0];
28         for i = 2:9
29             theta = 40*(i-1)+(randn(1)-0.5)*3;
30             r = 100+randn(1)*2;
31             data_test(i,:) = [r*cosd(theta) r*sind(theta)];
32         end
33         data = data_test;
34         [~,steps,~] = adjust_of_drone(data,data_std,learning_rate);
35         sums = sums+steps;
36     end
37     sums = sums/100;
38     dddata = [dddata;learning_rate sums];
39 end
40 plot(dddata(:,1),dddata(:,2),'-',LineWidth=1)
41 xlabel('learning rate')
```

```

42     ylabel('average iteration steps')
43
44     % learning_rate = 0.9;
45     %
46     % % 作图
47     % sz = 15;
48     % c1 = [0 0.4470 0.7410];
49     % c2 = [0.8500 0.3250 0.0980];
50     % [xc0,yc0,R0] = fit_circular(data);
51     % alpha0=0:pi/30:2*pi;
52     % x0=R0*cos(alpha0)+xc0;
53     % y0=R0*sin(alpha0)+yc0;
54     % plot(x0,y0,'-',LineWidth=1)
55     % axis equal
56     % hold on
57     %
58     % learning_rate = 1;
59     % [data_adjusted,steps,los] = adjust_of_drone(data,data_std,learning_rate);
60     % [xc,yc,R] = fit_circular(data_adjusted);
61     % alpha=0:pi/30:2*pi;
62     % x=R*cos(alpha)+xc;
63     % y=R*sin(alpha)+yc;
64     % plot(x,y,'-',LineWidth=1)
65     % hold on
66     %
67     % scatter(data(:,1),data(:,2),sz,c1,'filled');
68     % scatter(data_adjusted(:,1),data_adjusted(:,2),sz,c2,'filled');
69     % legend('originally fitted circle','eventually fitted circle','original data','adjusted data
70     %         ')
71     % xlabel(['iteration steps=',num2str(steps),' learning rate=',num2str(learning_rate),'
72     %         residual=',num2str(los)])
73
74     %函数
75     function indexes = choice_of_drone(data,data_std)
76     % 随机共三架无人机得到的角度坐标，用100的数据得到估计坐标m
77     % 再拟合成圆，最后计算估计值距离圆最近的三个无人机作为发射源
78     sample = randperm(9);
79     index = sample(1:2);
80     index = sort(index,"ascend");
81
82     % 所有飞机和当前选到的两架圆弧上的无人机以及圆心无人机的角
83     [alpha,clockwise] = get_alpha(data,index);
84     % 通过式子解出所有的坐标
85     coordinates = get_coordinates(index,alpha,clockwise,data_std);
86     % 拟合成圆形
87     coordinates_exclude = [];
88     for i = 1:9
89         if i == index(1) || i == index(2)
90             continue
91         end
92         coordinates_exclude = [coordinates_exclude;coordinates(i,:) i];
93     end
94     [xc,yc,R] = fit_circular(coordinates_exclude); % 中不包含发射的那几架飞机，但是这些所谓的坐标是，但是
95     % 其实都是假设的错误坐标coordinatesdata_std

```

```

94     sum = 0;
95     for i = 1:7
96         tmp = coordinates_exclude(i,:) - [xc, yc, 0];
97         sum = sum + abs(norm(tmp(:, 1:2)) - R);
98     end
99     if sum <= 10
100         indexes = sort(datasample(coordinates_exclude(:, 3), 3));
101         % 或者求距离最小的三个飞机
102         % error = [];
103         % for i = 1:9
104         %     error = [error; abs(norm(coordinates_exclude(i, 1:2) - [xc, yc]) - R)];
105         % end
106         % m=zeros(1,3);
107         % error_min=zeros(1,3);
108         % m(1)=find(error==min(error));
109         % error_min(1)=error(m(1));
110         % error(m(1))=max(error);
111         % m(2)=find(error==min(error));
112         % error_min(2)=error(m(2));
113         % error(m(2))=max(error);
114         % m(3)=find(error==min(error));
115         % error_min(3)=error(m(3));
116         % error(m(3))=max(error);
117         %
118         % % 发射信号飞机的编号
119         % indexes = sort([m(1);m(2);m(3)], "ascend");
120     else
121         indexes = sort([index datasample(coordinates_exclude(:, 3), 1)]);
122     end
123
124
125 end
126
127 function [data_adjusted, steps, los] = adjust_of_drone(data, data_std, learning_rate)
128 % 每次调整，通过三四架飞机来确定当前飞机的位置坐标/
129 % 再用估计坐标调整到标准位置上
130 data_adjusted = data;
131 % indexes = choice_of_drone(data, data_std);
132 indexes = sort(randperm(9));
133 steps = 0;
134 while loss(data_adjusted, data_std) >= 1e-2
135     % for i = 1:20000
136         steps = steps + 1;
137         [alpha_case12, clockwise12] = get_alpha(data_adjusted, [indexes(1), indexes(2)]);
138         [alpha_case13, clockwise13] = get_alpha(data_adjusted, [indexes(1), indexes(3)]);
139         coordinates12 = get_coordinates([indexes(1), indexes(2)], alpha_case12, clockwise12, data_std)
140         ;
141         coordinates13 = get_coordinates([indexes(1), indexes(3)], alpha_case13, clockwise13, data_std)
142         ;
143         coordinates = (coordinates13 + coordinates12) / 2;
144         data_adjusted = data_adjusted - learning_rate * (coordinates - data_std);
145         %     indexes = choice_of_drone(data_adjusted, data_std);
146         indexes = sort(randperm(9));
147     end
148     los = loss(data_adjusted, data_std);

```

```

147     end
148
149     function l = loss(data_adjusted,data_std)
150     tmp = (data_adjusted-data_std).^2;
151     l = sum(sqrt(tmp(:,1)+tmp(:,2)));
152     end
153
154     function [xc,yc,R] = fit_circular(coordinates)
155     % 通过参数拟合成圆形，得到圆心和半径值
156     x = coordinates(:,1); y = coordinates(:,2);
157     n=length(x); xx=x.*x; yy=y.*y; xy=x.*y;
158     A=[sum(x) sum(y) n;sum(xy) sum(yy)...
159       sum(y);sum(xx) sum(xy) sum(x)];
160     B=[-sum(xx+yy) ; -sum(xx.*y+yy.*y) ; -sum(xx.*x+xy.*y)];
161     a=A\B;
162     xc = -.5*a(1);
163     yc = -.5*a(2);
164     R = sqrt((a(1)^2+a(2)^2)/4-a(3));
165     end
166
167     function [angle,clockwise] = get_angle(coordinates)
168     % 通过三点坐标得到夹角角度大小
169     % 输入三点坐标，输出与的夹角1323
170     sub13 = coordinates(1,:)-coordinates(3,:);
171     k1 = sub13(2)/sub13(1);
172     sub23 = coordinates(2,:)-coordinates(3,:);
173     k2 = sub23(2)/sub23(1);
174     if sub13(1) == 0
175         if sub13(2)>=0 && sub23(1)>0 || sub13(2)<=0 && sub23(1)<0
176             clockwise = 1;
177         else
178             clockwise = -1;
179         end
180         angle = 90-atan(abs(k2))/pi*180;
181     elseif sub23(1) == 0
182         if sub13(1)>0
183             clockwise = -1;
184         else
185             clockwise = 1;
186         end
187         angle = 90-atan(abs(k1))/pi*180;
188     else
189         tmp = (k1-k2)/(1+k1*k2);
190         % 顺时针判断：到是顺时针，1323clockwise，否则为=1-1
191         if tmp>=0
192             clockwise = 1;
193         else
194             clockwise = -1;
195         end
196         angle = (atan(abs(tmp)))/pi*180;
197     end
198     end
199
200     function [alpha,clockwise] = get_alpha(data,index)
201     % 所有飞机和当前选到的两架圆弧上的无人机以及圆心无人机的夹角，第，项是第架飞机的ijialpha_j

```

```

202 % 是被选飞机index
203 % 只看相对于飞机的顺时针就可以直接判断, 和分别代表第一二架飞机的编号clockwiseindex1index2
204 alpha = []; clockwise = [];
205 for i = 1:9
206     if i == index(1)
207         [alpha2,clockwise23] = get_angle([0 0;data(index(2),:);data(i,:)]);
208         alpha = [alpha;0 alpha2 0];
209         clockwise13 = 1;
210         clockwise = [clockwise;clockwise13 clockwise23];
211     elseif i == index(2)
212         [alpha1,clockwise13] = get_angle([0 0;data(index(1),:);data(i,:)]);
213         alpha = [alpha;alpha1 0 0];
214         clockwise23 = 1;
215         clockwise = [clockwise;clockwise13 clockwise23];
216     else
217         [alpha1,clockwise13] = get_angle([0 0;data(index(1),:);data(i,:)]); % 原点和第一架
218         [alpha2,clockwise23] = get_angle([0 0;data(index(2),:);data(i,:)]); % 原点和第二架
219         [alpha3,~] = get_angle([data(index(1),:);data(index(2),:);data(i,:)]); % 两架圆弧
220         alpha = [alpha;alpha1 alpha2 alpha3];
221         clockwise = [clockwise;clockwise13 clockwise23];
222     end
223 end
224 end
225
226 function coordinates = get_coordinates(index,alpha,clockwise,data_std)
227 % 通过输入角度求出坐标
228 i = -clockwise;
229 coordinates = [];
230 % 第k个飞机的两个, 以及对应的角度kalpha
231 for k = 1:9
232     if k == index(1) || k == index(2)
233         coordinates = [coordinates;data_std(k,:)];
234     else
235         theta = (index(2)-index(1))*40;
236         tan_theta_hat = ((sind(alpha(k,1))*i(k,2)*sind(i(k,2)*alpha(k,2)-theta)-sind(alpha(k,2))*sind(alpha(k,1)))/ ...
237             (sind(alpha(k,2))*i(k,1)*cosd(alpha(k,1))-sind(alpha(k,1))*i(k,2)*cosd(i(k,2)*alpha(k,2)-theta)));
238         if i(k,1) >= 0
239             if tan_theta_hat>=0
240                 theta_hat = atan(tan_theta_hat)/pi*180;
241             else
242                 theta_hat = atan(tan_theta_hat)/pi*180+180;
243             end
244         else
245             if tan_theta_hat>=0
246                 theta_hat = atan(tan_theta_hat)/pi*180+180;
247             else
248                 theta_hat = 360+atan(tan_theta_hat)/pi*180;
249             end
250         end
251         theta_hat = theta_hat+(index(1)-1)*40;
252         r = 100*i(k,1)*sind(theta_hat+i(k,1)*alpha(k,1))/sind(alpha(k,1));
253         coordinates = [coordinates;r*cosd(theta_hat) r*sind(theta_hat)];
254     end
end

```

255
256

end
end