

基于纯方位无源定位的无人机三角定位模型

摘 要

本文围绕无人机在遂行编队飞行中的纯方位无源定位问题，建立了被动接收信号无人机的三角定位模型，并给出了如何将存在位置偏差的队列调整为无位置偏差的队列的方案。

问题 1 第（1）问中，我们实现了通过三架位置已知的无人机对任意一架无人机的定位。首先通过构造待测点与已知点的三角形及其外接圆，建立三角定位模型，其次寻找几何关系确定外接圆的圆心和半径，最后，由于待测无人机同时位于多个外接圆上，可将其代入已经确定的外接圆方程中，构建方程组，利用最小二乘法对方程组进行求解，即可得到待测无人机的位置，具体结果见文中表达式。

问题 1 第（2）问中，我们利用假设法确定了在两架位置已知的无人机的情况下，还需几架位置未知的无人机发射信号来实现任一无人机的有效定位。首先假设只需 1 架，按照第（1）问建立方程组，方程组缺少已知量无法求解，从而不能实现无人机的有效定位，假设不成立；其次假设还需 2 架无人机，同样建立方程组，构建出未知量之间的关系，发现方程组已知量足够可以求解，从而实现无人机的有效定位，假设成立。故最后得出：除 $FY00, FY01$ 外，还需要两架无人机发射信号，才能实现无人机的有效定位的结论。

问题 1 第（3）问，我们通过计算无人机初始位置与标准位置的欧氏距离，选取欧氏距离最小的无人机和编号为 0, 1 的无人机建立三角定位模型，并结合贪婪算法的基本思想，给出了无人机调至标准位置的调整方案。通过 7 次定位和调整得到了最终的调整结果，具体调整方案和调整结果结果见文中图 14，并对调整结果进行误差分析，证明了调整方案的可行性，且调整结果准确度高。

针对问题 2，我们给出了将存在位置偏差的队列调整为无位置偏差的标准锥形队列的调整方案。首先假设锥形顶端的三架无人机位于无偏差位置，对其建立三角定位模型，则可确定出一个存在位置偏差的无人机的坐标；其次，无人机根据坐标调整至标准位置，并从接收信号转变为发射信号，参加到下一个点的定位，以此类推，直至所有存在偏差的点都经过 1 次调整时调整结束，具体调整方案见文中描述。最后，我们对该方案进行计算机仿真和误差分析，证明了该调整方案具有可行性，且调整结果准确度高。

关键词：三角定位模型 最小二乘法 贪婪算法 计算机仿真

一、问题重述

1.1 问题背景

无人机集群在遂行编队飞行时，为避免外界干扰，应尽可能保持电磁静默，少向外发射电磁波信号。故为了保持队形，采用纯方位无源定位的方法调整无人机的位置，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。

1.2 问题提出

已知编队中每架无人机均有固定编号，且在编队中与其他无人机的相对位置关系保持不变，同时接受信号的无人机所接收到的方向信息约定为：该无人机与任意两架发射信号无人机连线之间的夹角，我们需建立数学模型解决以下问题：

问题 1: 编队由 10 架无人机组成，组成圆形编队，其中 9 架无人机（编号 FY00~FY09）均匀分布在某一圆周上，另一架无人机（编号 FY00）位于圆心。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。

（1）位于圆心的无人机（FY00）和编队中另两架无人机发射信号，其余位置略有偏差的无人机接收信号。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，确定还需几架无人机发射信号，才能实现无人机的有效定位。

（3）按编队要求，1 架无人机位于圆心，令 9 架无人机均匀分布在半径为 100 米的圆周上。当初始时刻无人机的位置略有偏差时，请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（每次调整的时间忽略不计），使得 9 架无人机最终均匀分布在某个圆周上，无人机初始位置见表 1。

问题 2: 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（见图 3，直线上相邻两架无人机的间距相等，如 50 米）。仍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

2.1 问题 1 第（1）小问的分析

问题定义了无位置偏差的圆形编队的排列规则，并给定了无位置偏差的 FY00 和任意两架编号已知的无人机用于发射信号，要求我们据此确定任意一架编号未知且存在位置偏差的无人机位置。根据排列规则我们可以知道：若一架无人机编号已知且位置无偏差，即可知道它的位置，所以问题 1 第（1）小问的实质为用三个已知点确定一个未知点，故我们可以通过建立三角定位模型进行求解。

2.2 问题 1 第（2）小问的分析

该问给出：用于发射信号的无人机为编号 FY00、FY01 和若干架编号未知的无人机，且所有用于反射信号的无人机均无位置偏差。我们需据此确定除 FY00 和 FY01 外，还需几架编号未知的无人机发射信号，才能实现任意一架无人机的有效定位。问题虽

为确定无人机数量，但其本质仍为定位问题，故可以基于第（1）小问建立的三角模型进行求解。

2.3 问题 1 第（3）小问的分析

题目给定了一个除 $FY00$ 和 $FY01$ 外，均存在位置偏差的队列，我们需要给出合理的调整方案，使其能够调整为无位置偏差的圆形队列（半径为 100m ），且该问下，所有无人机的编号已知。由问题 1 第（1）小问可知，要想定位一个存在位置偏差的点，至少需要三个位置已知的点，故我们可以先从存在位置偏差的点中选取一个最接近标准位置的点，将其视为无位置偏差的点（即位置已知的点），再结合已知的 $FY00$ 和 $FY01$ ，即可构建三角定位模型，并依据此进行第一次调整。再从调整过后的点中选出最接近标准位置的点，以此类推，直到所有存在位置偏差的点都至少有一次被视为标准点即完成位置的调整，可借助贪婪算法进行求解。

2.4 问题 2 的分析

问题 2 要求我们给出锥形编队的调整方案，由问题 1 第（1）小问可知，要想定位一个存在位置偏差的点，至少需要三个位置已知的点。故我们约定锥形队列的顶点和与它相邻的两个点不存在位置偏差（即位置已知），再结合问题 1 第（1）小问建立的三角定位模型，通过多次调整，即可将其调整为标准的锥形队列。

三、模型假设

1. 无人机的位置偏差只考虑水平方向的偏差，即所有无人机处于同一高度；
2. 假设无人机可以看成质点；
3. 在没有具体说明无偏差圆形编队的半径大小时，默认为半径已知，记作 R ；
4. 圆形编队中，均以 $FY00$ 为圆心，圆周上的无人机按照 $FY01 \sim FY09$ 逆时针顺序排布（即不会出现 $FY01$ 、 $FY03 \dots$ 类似不按顺序排的情况）；
5. 假设接收到的信号不受外界任何干扰；
6. 所有无人机均能正常工作。

四、符号说明

符号	说明
$FY0k, FY0j$	圆形编队中无位置偏差且编号已知的任意 2 架用于发射信号的无人机编号
A	被动接收信号的待测无人机
θ	任意两架无位置偏差且编号已知的无人机之间的夹角
(x_i, y_i)	构建的外接圆圆心
r_i	构建的外接圆的半径
(x, y)	待测无人机 A 的位置坐标

五、模型的建立与求解

5.1 问题 1 第 (1) 问模型的建立与求解

根据题目可知：无偏差的圆形编队为 9 架无人机 ($FY01 \sim FY09$) 均匀分布在某一圆周上，一架无人机 ($FY00$) 位于圆心，则圆周上任意相邻两架无人机之间的夹角为固定值 40° 。

第 (1) 小问中，题目指明共有三架无人机发射信号，分别为 $FY00$ 和任意两个编号已知且位置无偏差的无人机 (为叙述方便，将该两架无人机分别记为 $FY0k, FY0j$ ， k, j 为 $1 \sim 9$ 的任意整数，且 $k \neq j$)，同时，还存在位置有偏差且编号未知的待定位无人机，记为点 A 。

设无偏差的圆形编队为以 $FY00$ 为圆心，半径为 R 的圆周，则第 (1) 小问的示意图如下：

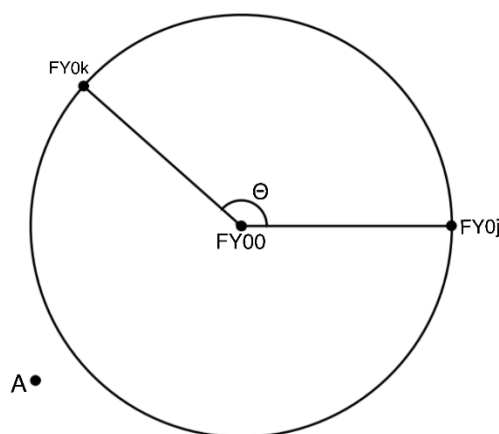


图 1 问题 1 第 (1) 问示意图

其中， θ 为 $FY0k, FY0j$ 之间的夹角，其大小为 (由于这两架无人机编号已知，即 k, j 为一个确定的数，所以计算出的 θ 也为一个确定的数)：

$$\theta = |k - j| \times 40^\circ \quad (1)$$

记 $FY00$ 为点 O , $FY0j$ 为点 B , $FY0k$ 为点 C ，令点 O 为原点， OB 为 x 轴，垂直于 OB 为 y 轴建立直角坐标系，如图 2 所示：

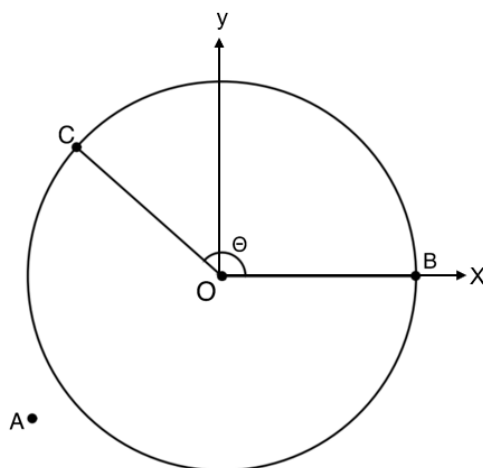


图 2 问题 1 第 (1) 问坐标系

为了能够求出 A 的具体坐标，我们可以通过寻找图(2)中的几何关系进行位置确定，但由于无人机接收到的只有角度信息，致使图(2)中所含信息太少，所以无法直接通过寻找几何关系来确定 A 的位置。对此，我们可以围绕 A 建立多个方程，使得 A 同时满足建立的方程，即可求出 A 的位置。

连接 OA, OB, AC, BC ，得到 $\triangle AOB$ ， $\triangle AOC$ ， $\triangle ABC$ ，分别做三个三角形的外接圆，记为 $\odot O_1$ ， $\odot O_2$ ， $\odot O_3$ ，如图3所示：

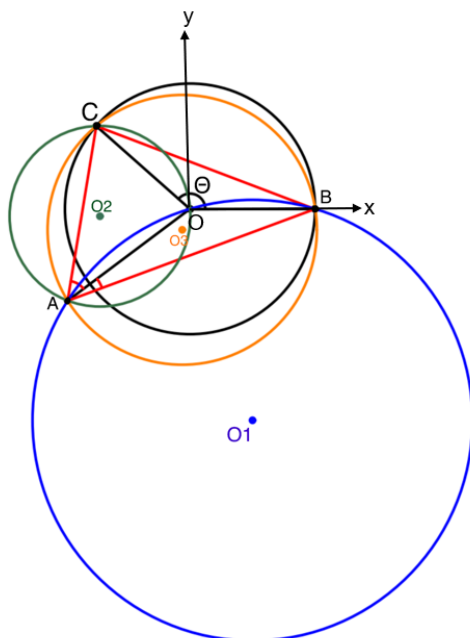


图3 问题1第(1)问外接圆示意图

连接并延长 OO_1 交 $\odot O_1$ 与点 D ，连接 BD ；连接并延长 OO_2 交 $\odot O_2$ 与点 E ，连接 BE ；连接并延长 BO_3 交 $\odot O_3$ 于点 F ，连接 CF ，则有 $\triangle BOD$ 、 $\triangle COE$ 、 $\triangle BCF$ 和直径 OD 、 OE 、 BF ，如图4所示：

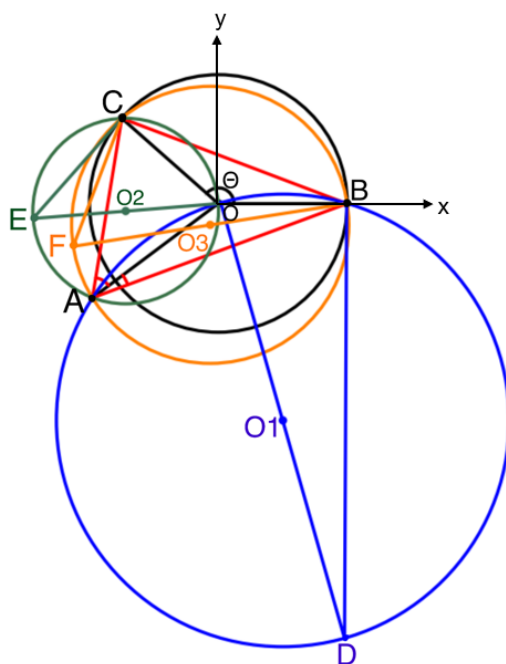


图4 问题1第(1)问三角定位模型示意图

图 4 即为用于定位的三角定位模型，此时的 A 点位于 $\odot O_1$ ， $\odot O_2$ 和 $\odot O_3$ 的交点上，即 A 点同时满足三个圆的方程。据此可以求解出被动接收信号无人机 A 的位置，具体过程如下：

首先，对图 4 中的已知量（此处的已知量指的是可根据题干信息直接获取到的信息，无需另作证明）进行整理，如表 1：

表 1 问题 1 第 (1) 问三角定位模型已知量概览表

符号	含义	大小
BO 、 CO	无位置偏差的圆形编队的半径	记为 R
$\angle BAO$	无人机 A 接收到的位置信息	记为 $\angle \alpha_1$
$\angle CAO$	无人机 A 接收到的位置信息	记为 $\angle \alpha_2$
$\angle CAB$	无人机 A 接收到的位置信息	记为 $\angle \alpha_3$
$\angle BOC$	任意两架无位置偏差且编号已知的无人机之间的夹角	记为 $\angle \theta$ ，具体表达式见式 (1)

则可得到 B 、 C 两点的坐标，分别为： $B(R, 0)$ ； $C(R \cos \theta, R \sin \theta)$ 。

其次，对图 4 中的未知量进行设定：

- (1) 设 $\odot O_1$ 圆心坐标为 (x_1, y_1) ，半径为 r_1 ；
- (2) 设 $\odot O_2$ 圆心坐标为 (x_2, y_2) ，半径为 r_2 ；
- (3) 设 $\odot O_3$ 圆心坐标为 (x_3, y_3) ，半径为 r_3 ；
- (4) 设待测无人机 A 的坐标为 (x, y)

最后，结合 $\triangle BOD$ ， $\triangle COE$ ， $\triangle BCF$ 建立相关方程组对 A 点位置进行求解。

• 针对 $\triangle BOD$ ：

由图 4 可知， $\angle BDO$ 与 $\angle \alpha_1$ 为同一圆上 ($\odot O_1$)，同一弧长 (\widehat{BO}) 所对应的不同圆周角，根据圆周角的性质^[2]：同一圆中，同弧所对应的圆周角大小相等，则有：

$$\angle BDO = \angle \alpha_1 \quad (2)$$

同时，坐标原点 $O(0, 0)$ 和 $B(R, 0)$ 位于 $\odot O_1$ ，根据圆的标准方程：

$$(x - a)^2 + (y - b)^2 = r^2 \quad (3)$$

(x, y) 为圆上任意一点， (a, b) 为圆心， r 为圆的半径。则可针对坐标原点 $O(0, 0)$ 列写方程：

$$(x_1 - 0)^2 + (y_1 - 0)^2 = r_1^2 \quad (4)$$

针对 $B(R, 0)$ 列写方程：

$$(x_1 - R)^2 + (y_1 - 0)^2 = r_1^2 \quad (5)$$

并且， $\angle OBD$ 为 $\odot O_1$ 直径 OD 所对的圆周角，根据直径所对的圆周角为 90° 可知： $\angle OBD = 90^\circ$ ，即 $\triangle BOD$ 为直角三角形。所以在 $\triangle BOD$ 中有：

$$\frac{BC}{OD} = \frac{R}{2r_1} = \sin \angle BDO = \sin \angle \alpha_1 \quad (6)$$

根据二倍角公式，可将式(6)整理为：

$$R^2 = 2r_1^2(1 - \cos 2\alpha_1) \quad (7)$$

综上，针对 $\triangle BOD$ 有以下方程组：

$$\begin{cases} x_1^2 + y_1^2 = r_1^2 \\ (x_1 - R)^2 + y_1^2 = r_1^2 \\ R^2 = 2r_1^2(1 - \cos 2\alpha_1) \end{cases} \quad (8)$$

该方程组为普通的三元二次方程组，通过简单的代入即可求出解，具体解如下：

$$\begin{cases} x_1^2 = \frac{R^2}{4} \\ y_1^2 = \frac{R^2}{2(1 - \cos 2\alpha_1)} - \frac{R^2}{4} \\ r_1^2 = \frac{R^2}{2(1 - \cos 2\alpha_1)} \end{cases} \quad (9)$$

• 针对 $\triangle COE$ ：

同理，在 $\triangle COE$ 中， $\angle CEO$ 与 $\angle \alpha_2$ 为 $\odot O_2$ 中同弧所对的不同圆周角； $\angle ECO$ 为直径 OE 所对的圆周角；点 C 和点 O 均在 $\odot O_2$ 上。同 $\triangle BOD$ 的一样，可对 $\triangle COE$ 列如下方程组（由于每个方程的得来原理与 $\triangle BOD$ 中的各方程的得来原理相同，为避免叙述累赘，故不再对 $\triangle COE$ 中的各方程由来做具体叙述）：

$$\begin{cases} x_2^2 + y_2^2 = r_2^2 \\ (x_2 - R \cos \theta)^2 + (y_2 - R \sin \theta)^2 = r_2^2 \\ R^2 = 2r_2^2(1 - \cos 2\alpha_2) \end{cases} \quad (10)$$

解得：

$$\begin{cases} x_2 = \frac{R}{2\cos\theta(1+\tan^2\theta)} \left(\frac{\sin^2\theta \tan^2\alpha_2 + \sin^2\alpha_2 - \sin^2\alpha_2 \tan^2\alpha_2}{\sin^2\alpha_2 \sin^2\theta \tan^2\alpha_2} \right) \\ y_2 = \frac{R}{2\sin\theta} - \frac{R}{2\sin\theta(1+\tan^2\theta)} \left(\frac{\sin^2\theta \tan^2\alpha_2 + \sin^2\alpha_2 - \sin^2\alpha_2 \tan^2\alpha_2}{\sin^2\alpha_2 \sin^2\theta \tan^2\alpha_2} \right) \\ r_2^2 = \frac{R^2}{2(1 - \cos 2\alpha_2)} \end{cases} \quad (11)$$

• 针对 $\triangle BCF$ ：

同理，在 $\triangle BCF$ 中， $\angle BFC$ 与 $\angle \alpha_3$ 为 $\odot O_3$ 中同弧所对的不同圆周角； $\angle BCF$ 为直径 BF 所对的圆周角；点 B 和点 C 均在 $\odot O_3$ 上。同 $\triangle BOD$ 的一样，可对 $\triangle BCF$ 列如下方程组（由于每个方程的得来原理与 $\triangle BOD$ 中的各方程的得来原理相同，为避免叙述累赘，故不再对 $\triangle BCF$ 中的各方程由来做具体叙述）：

$$\begin{cases} (x_3 - R)^2 + y_3^2 = r_3^2 \\ (x_3 - R \cos \theta)^2 + (y_3 - R \sin \theta)^2 = r_3^2 \\ (R - R \cos \theta)^2 + (R \sin \theta)^2 = 2r_3^2(1 - \cos 2\alpha_3) \end{cases} \quad (12)$$

解得：

$$\begin{cases} x_3 = \frac{\sin^2 \theta}{\sin^2 \theta + (1 - \cos \theta)^2} \left\{ R + 2 \left[r_3^2 - \frac{R^2(1 - \cos \theta)^2}{\sin^2 \theta} + \frac{r_3^2(1 + \cos \theta)^2}{\sin^2 \theta} \right] \right\} \\ y_3 = \frac{(1 + \cos \theta) \sin^2 \theta}{\sin \theta [\sin^2 \theta + (1 - \cos \theta)^2]} \left\{ R + 2 \left[r_3^2 - \frac{R^2(1 - \cos \theta)^2}{\sin^2 \theta} + \frac{r_3^2(1 + \cos \theta)^2}{\sin^2 \theta} \right] \right\} \\ r_3^2 = \frac{R^2(1 - \cos \theta)}{1 - \cos 2\alpha_3} \end{cases} \quad (13)$$

至此， $\odot O_1$ 、 $\odot O_2$ 、 $\odot O_3$ 的圆心和半径均已确定，所以三个圆的标准方程已经确定。由于在构建该三个外接圆时，均用到了待测无人机的位置 A 点 (x, y) ，即 A 点同时位于3个圆上，故 A 需同时满足三个圆的方程，则针对 A 可列下列方程组：

$$\begin{cases} (x_1 - x)^2 + (y_1 - y)^2 = r_1^2 \\ (x_2 - x)^2 + (y_2 - y)^2 = r_2^2 \\ (x_3 - x)^2 + (y_3 - y)^2 = r_3^2 \end{cases} \quad (14)$$

将方程组(14)各方程展开移项之后，整理得到以下方程组：

$$\begin{cases} r_1^2 - (x_1^2 + y_1^2) = (x^2 + y^2) - 2xx_1 - 2yy_1 \\ r_2^2 - (x_2^2 + y_2^2) = (x^2 + y^2) - 2xx_2 - 2yy_2 \\ r_3^2 - (x_3^2 + y_3^2) = (x^2 + y^2) - 2xx_3 - 2yy_3 \end{cases} \quad (15)$$

令 $K_i = x_i^2 + y_i^2$ ($i = 1, 2, 3$), $L = x^2 + y^2$ ，则方程组(15)可以写成矩阵的形式：

$$\begin{bmatrix} r_1^2 - K_1 \\ r_2^2 - K_2 \\ r_3^2 - K_3 \end{bmatrix} = \begin{bmatrix} -2x_1 & -2y_1 & 1 \\ -2x_2 & -2y_2 & 1 \\ -2x_3 & -2y_3 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ L \end{bmatrix} \quad (16)$$

则可将上述方程组记为：

$$Y = AX \quad (17)$$

利用最小二乘法即可求出：

$$X = (A^T A)^{-1} A^T Y \quad (18)$$

从而可以得到被动接收信号无人机位置的坐标 (x, y) ，继而实现了无人机的有效定位。

至此，问题1第(1)小问模型建立求解结束。

5.2 问题1第(2)问模型的建立与求解

对于第(2)小问，用于发射信号的无人机为编号 $FY00$ 、 $FY01$ 和若干架编号未知的无人机，且所有用于反射信号的无人机均无位置偏差。我们需据此确定除 $FY00$ 和 $FY01$ 外，还需几架编号未知的无人机发射信号，才能实现任意一架无人机的有效定位。

虽然除 $FY00$ 、 $FY01$ 外，其余发射信号的无人机编号未知，但由于其位置无偏差，根据圆形编队的要求，编号未知的无人机位置只可能出现在下图所示的红色点上（任意两个相邻红点之间的夹角为 40° ）：

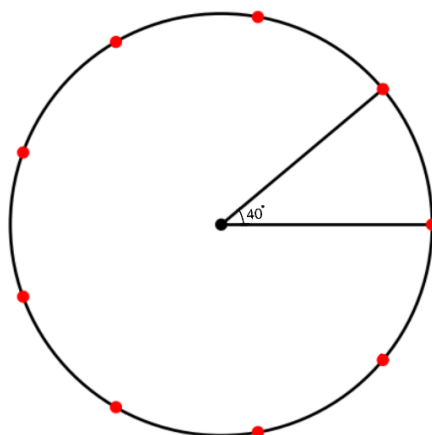


图 5 编号未知的无人机可能处于的位置示意图

第（2）小问虽然需要求解的是发射信号的无人机数量，但其本质上还是定位问题，我们以 $FY00$ 为坐标原点 O ， O - $FY01$ 为 x 轴，垂直于它射线的作为 y 轴建立直角坐标系，仍然可以基于三角定位模型对该问进行求解，

• 假设 1：除 $FY00$ 、 $FY01$ 外，还需 1 架无人机即可实现有效定位

将该编号未知无人机记为 $FY0k_1$ （ k_1 为 2~9 的任意一个整数），待测点记为 A ，依照第（1）小问，可对该情况下建立三角定位模型（由于三角定位模型建立方法和建立过程相同，均为绘制以“1 个待测点—2 个发射信号点”作为顶点的三角形，再对构建出的三角形绘制外接圆进行模型建立。故为避免叙述累赘，该问不再给出该问三角定位模型建立的过程，只给出最终的三角定位模型图），如图 5：

其中， θ 为 $FY0k_1$ 与 $FY01$ 之间的夹角，其大小为：

$$\theta = (k_1 - 1) \times 40^\circ \quad (19)$$

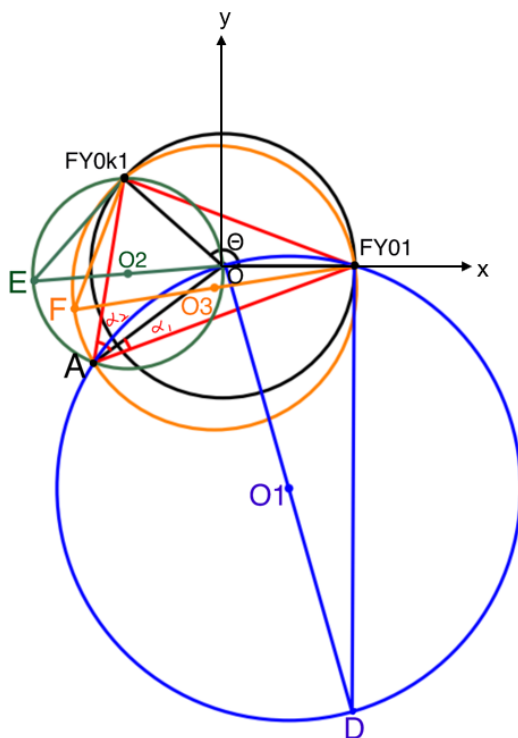


图 6 假设 1 的三角定位模型示意图

根据图 5 我们可以看出，假设 1 建立的模型图于第（1）小问相同，故建立的方程组形式也应相同。但与第（1）小问不同的是，此时的编号未知，即 k_1 为一个不确定的数，所以 θ 也为一个未知的数，无法作为已知量代入方程进行求解，使得建立的方程组缺少已知量无法进行求解，不能实现无人机的定位，与假设 1 矛盾。所以只添加一架无人机发射信号无法实现无人机的有效定位，对此我们提出假设 2。

• 假设 2：除 $FY00$ 、 $FY01$ 外，还需 2 架无人机即可实现有效定位

将添加的两架无人机记为 $FY0k_1$ 、 $FY0k_2$ (k_1 、 k_2 为 2~9 的任意整数，且 $k_1 \neq k_2$ ， $k_2 > k_1$)，待测点记为 A （为叙述方便，下文将无人机 $FY00$ 、 $FY01$ 、 $FY0k_1$ 、 $FY0k_2$ 分别记为点 O 、点 A 、点 B 、点 C 、点 D ），此时增加了一架无人机，能与待测点构成的三角形数量增多，如图 7：

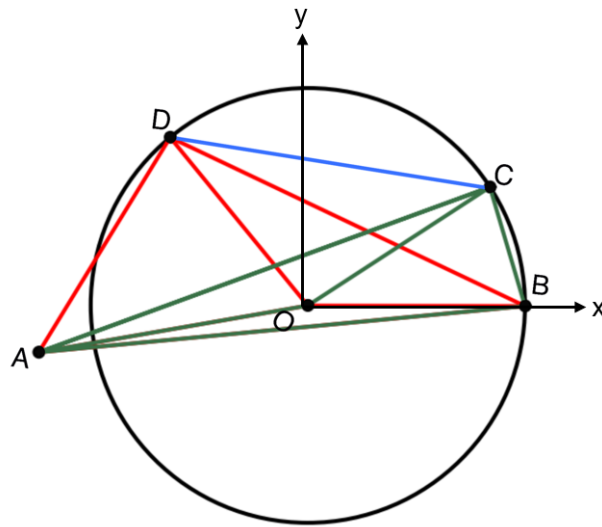


图 7 假设 2 情况下可构造的三角形示意图

从图中可以看出，以 A 为顶点，任意两架无人机为余下两个顶点构造的三角形共有 6 个，即可构造 6 个圆的方程。但由第（1）小问可知，三个方程即可确定 A 的位置，故我们对构造出的三角形进行取舍。

为了使每一个发射信号的无人机都参与到定位中，选取的三角形中，其顶点应包含点 B 、 C 、 O 、 D ；同时由于原点坐标为 $(0,0)$ ，为了计算方便，我们应使选取的三角形其顶点尽可能多的位于原点。最终，我们选取 $\triangle AOD$ 、 $\triangle AOC$ 、 $\triangle AOB$ ，并对其构建外接圆，以此建立第（2）小问的三角定位模型（模型建立过程同第 1 小问），如图 8：

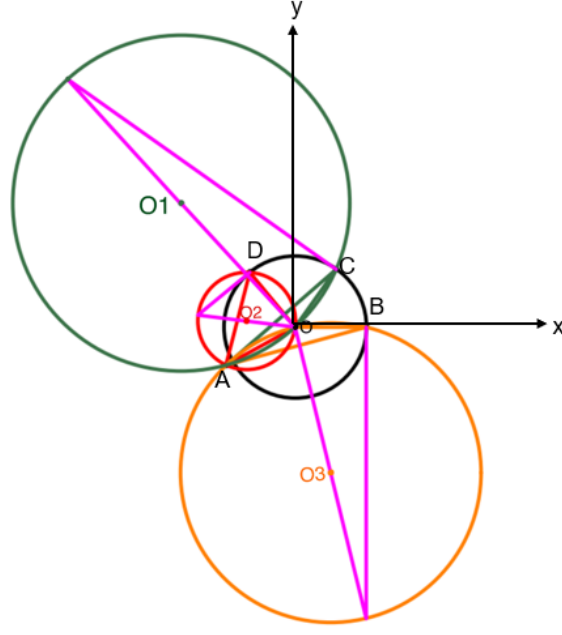


图 8 假设 2 的三角定位模型示意图

记 $\angle BOC$ 为 θ_1 ， $\angle DOB$ 为 θ_2 ，根据无偏差圆形编队要求可以得到：

$$\begin{cases} \theta_1 = (k_1 - 1) \times 40^\circ \\ \theta_2 = (k_2 - 1) \times 40^\circ \end{cases} \quad (20)$$

则可已得到 C 、 D 两点的坐标分别为： $C(R \cos \theta_1, R \cos \theta_1)$ ， $D(R \cos \theta_2, R \cos \theta_2)$ 。

但是由于该图涉及到的角度、三角形和圆形太多，若只使用一个图来完成所有方程建立，会使得阅读性降低，故我们将图 8 拆解为 3 部分，对每部分建立方程组。

部分 1：以 $\triangle AOC$ 构建的外界圆 $\odot O_1$ 部分，如图 9：

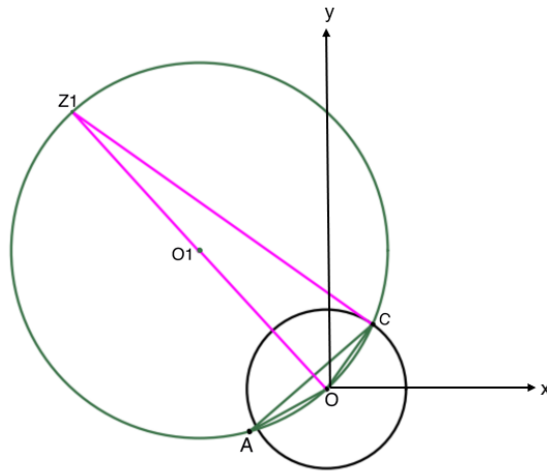


图 9 假设 2 的三角定位模型部分 1

由图可知， $\angle CZ_1O$ 与 $\angle CAO$ 为同弧所对的不同圆周角； $\angle Z_1CO$ 为直径所对的圆周角； $\angle CAO$ 为待测点 A 接收到无人机 C 、 O 发射信息的夹角，记为 $\angle \beta_1$ 。设 $\odot O_1$ 圆心坐标为 (x_1, y_1) ，半径为 r_1 ，同第 (1) 小问建立方程组的方法，可以建立该部分的方程组：

$$\begin{cases} (x_1 - 0)^2 + (y_1 - 0)^2 = r_1^2 \\ (x_1 - R \cos \theta_1)^2 + (y_1 - R \sin \theta_1)^2 = r_1^2 \\ R = 2r_1 \sin \beta_1 \end{cases} \quad (21)$$

解得：

$$\begin{cases} x_1 = \frac{R \cos \theta_1 + \sqrt{R^2 \left(\frac{\cos 2\theta_1 - 1}{2} \right) + 4 \sin^2 \theta_1 r_1^2}}{2} \\ y_1 = \frac{\frac{R}{2} - \frac{R \cos^2 \theta_1 + \cos \theta_1 \sqrt{R^2 \left(\frac{\cos 2\theta_1 - 1}{2} \right) + 4 \sin^2 \theta_1 r_1^2}}{2}}{\sin \theta_1} \\ r_1 = \frac{R}{2 \sin \beta_1} \end{cases} \quad (22)$$

部分 2：以 $\triangle AOD$ 构建的外界圆 $\odot O_2$ 部分，如图 10：

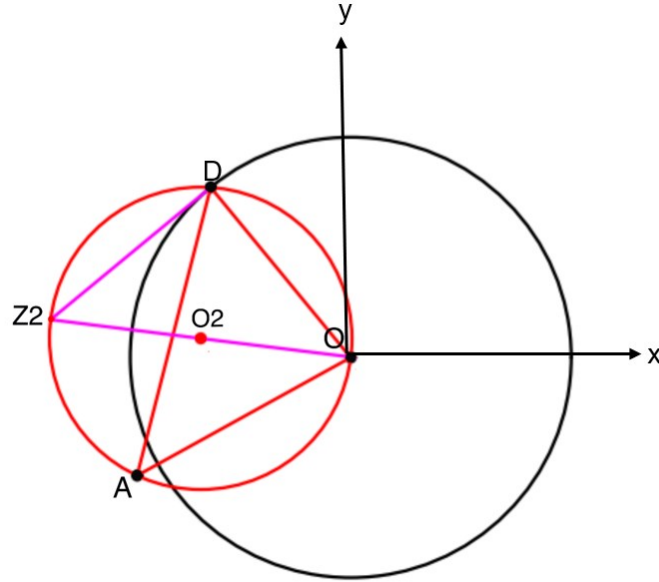


图 10 假设 2 的三角定位模型部分 2

由图可知， $\angle DZ_2O$ 与 $\angle DAO$ 为同弧所对的不同圆周角； $\angle Z_2DO$ 为直径所对的圆周角； $\angle DAO$ 为待测点 A 接收到无人机 D 、 O 发射信息的夹角，记为 $\angle \beta_2$ 。设 $\odot O_2$ 圆心坐标为 (x_2, y_2) ，半径为 r_2 ，该部分建立的方程组为：

$$\begin{cases} (x_2 - 0)^2 + (y_2 - 0)^2 = r_2^2 \\ (x_2 - R \cos \theta_2)^2 + (y_2 - R \sin \theta_2)^2 = r_2^2 \\ R = 2r_2 \sin \beta_2 \end{cases} \quad (23)$$

解得：

$$\left\{ \begin{array}{l} x_2 = \frac{R \cos \theta_2 + \sqrt{R^2 \left(\frac{\cos 2\theta_2 - 1}{2} \right) + 4 \sin^2 \theta_2 r_2^2}}{2} \\ y_2 = \frac{\frac{R}{2} - \frac{R \cos^2 \theta_2 + \cos \theta_2 \sqrt{R^2 \frac{(\cos 2\theta_2 - 1)}{2} + 4 \sin^2 \theta_2 r_2^2}}{2}}{\sin \theta_2} \\ r_2 = \frac{R}{2 \sin \beta_2} \end{array} \right. \quad (24)$$

部分 3: 以 $\triangle AOB$ 构建的外界圆 $\odot O_3$ 部分, 如图 11:

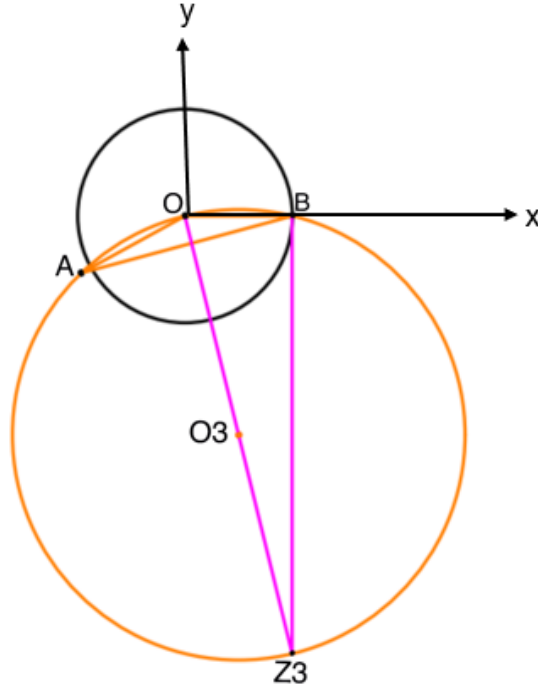


图 11 假设 2 的三角定位模型部分 3

由图可知, $\angle BZ_2O$ 与 $\angle DAO$ 为同弧所对的不同圆周角; $\angle Z_2BO$ 为直径所对的圆周角; $\angle BAO$ 为待测点 A 接收到无人机 B 、 O 发射信息的夹角, 记为 $\angle \beta_3$ 。设 $\odot O_3$ 圆心坐标为 (x_3, y_3) , 半径为 r_3 , 该部分建立的方程组为:

$$\left\{ \begin{array}{l} (x_3 - 0)^2 + (y_3 - 0)^2 = r_3^2 \\ (x_3 - R)^2 + (y_3 - 0)^2 = r_3^2 \\ R = 2r_3 \sin \beta_3 \end{array} \right. \quad (25)$$

解得:

$$\left\{ \begin{array}{l} x_3^2 = \frac{R^2}{4} \\ y_3^2 = r_3^2 - \frac{R^2}{4} \\ r_3^2 = \frac{R^2}{4 \sin^2 \beta_3} \end{array} \right. \quad (26)$$

此时求解出的 $\odot O_1$ 和 $\odot O_2$ 的圆心坐标含有 k_1, k_2 两个未知数, 对此我们还需找出 k_1, k_2 之间的关系式。对此, 我们连接 AD 、 CD 、 AC , 形成 $\triangle ADC$, 并构建它的外接圆 $\odot O_4$, 连接并延长 OO_4 交 $\odot O_4$ 于点 Z_4 , 连接 CZ_4 、 DZ_4 , 如图 12:

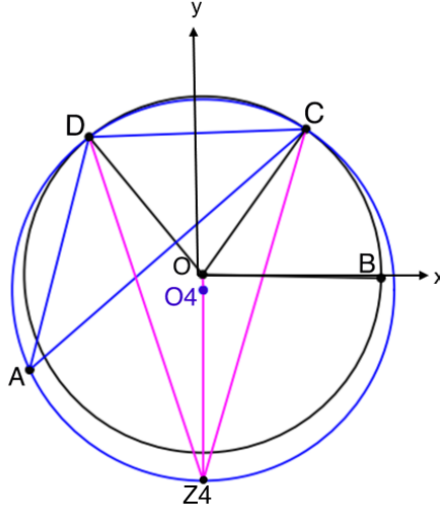


图 12 用于求解 k_1, k_2 关系式的示意图

由图可知, $\angle DAC$ 为待测点 A 接收到无人机 D 、 C 发射信息的夹角, 记为 $\angle\beta_4$; $\angle DZ_4C$ 与 $\angle DAC$ 为同弧所对的不同圆周角, 且 $\angle DZ_4C$ 与 $\angle DOC$ 分别为同弧所对的圆周角和圆心角, 且 $\angle DOC$ 满足如下关系式:

$$\angle DOC = \angle DOB - \angle COB = (k_2 - 1) \times 40^\circ - (k_1 - 1) \times 40^\circ = (k_2 - k_1) \times 40^\circ \quad (27)$$

根据同弧所对的不同圆周角相等, 且同弧所对的圆周角等于同弧所对的圆心角的一半, 则可得到 k_1, k_2 之间的关系式:

$$(k_2 - k_1) \times 40^\circ = 2\beta_4 \quad (28)$$

同时在 $\odot O_4$ 中, $\angle DAC$ 为待测点 A 接收到无人机 C 、 D 发射信息的夹角, 记为 $\angle\beta_4$ 。设 $\odot O_4$ 圆心坐标为 (x_4, y_4) , 半径为 r_4 , 则还可对该部分列写方程组:

$$\begin{cases} (x_4 - R \cos \theta_1)^2 + (y_4 - R \sin \theta_1)^2 = r_4^2 \\ (x_4 - R \cos \theta_2)^2 + (y_4 - R \sin \theta_2)^2 = r_4^2 \\ \frac{R \sin(\theta_2 - \theta_1)}{2 \sin \beta_4} = r_4 \end{cases} \quad (29)$$

解得:

$$\begin{cases} y_4 = \frac{\frac{2R \sin \theta_1 (\sin \theta_2 - \sin \theta_1)}{\cos \theta_1 - \cos \theta_2} + \left[\frac{2R \cos \theta_1 (\sin \theta_2 - \sin \theta_1)}{\cos \theta_1 - \cos \theta_2} \right]^2 - 4(R^2 - r_4^2) \left[\left(\frac{\sin \theta_2 - \sin \theta_1}{\cos \theta_1 - \cos \theta_2} \right) + 1 \right]}{2 \left(\frac{\sin \theta_2 - \sin \theta_1}{\cos \theta_1 - \cos \theta_2} \right)^2 + 2} \\ x_4 = \left(\frac{\sin \theta_2 - \sin \theta_1}{\cos \theta_1 - \cos \theta_2} \right) \times y_4 \\ r_4 = \frac{R \sin(\theta_2 - \theta_1)}{2 \sin \beta_4} \end{cases} \quad (30)$$

最后, 设待测点 A 的坐标为 (x, y) , 将其代入 $\odot O_1 \sim \odot O_4$ 四个圆的方程, 可以得到如下方程组:

$$\begin{cases} (x_1 - x)^2 + (y_1 - y)^2 = r_1^2 \\ (x_2 - x)^2 + (y_2 - y)^2 = r_2^2 \\ (x_3 - x)^2 + (y_3 - y)^2 = r_3^2 \\ (x_4 - x)^2 + (y_4 - y)^2 = r_4^2 \end{cases} \quad (31)$$

此时方程中出现的 k_1, k_2 可根据式(28)相消,则此时的方程组中不再含有多余未知量,通过最小二乘法即可求出点 A 的坐标(由于本问只需给出还需几架无人机,故不再对具体坐标进行求解),假设2成立。

所以除 $FY00$, $FY01$ 外,还需两架无人机来实现有效定位。

5.3 问题1第(3)问模型的建立与求解

该问给定10架无人机的初始位置,我们需制定方案将其调整为半径为100m且位置无偏差的圆形编队,即将下图中的蓝点(初始位置)向红点(标准位置)调整:

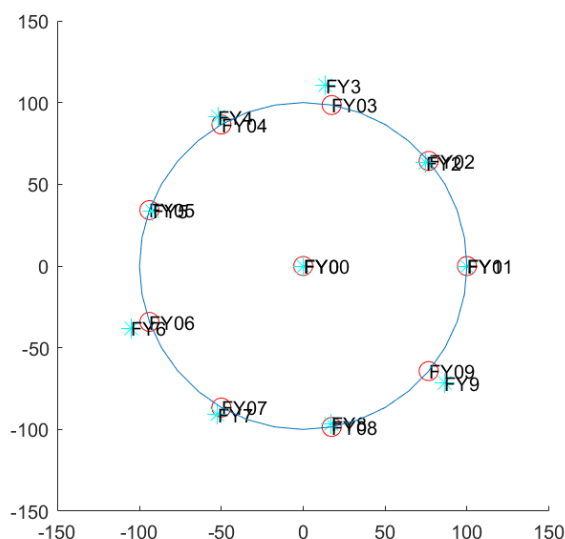


图13 问题1第(3)问初始位置于标准位置对比图

据题目可得:标准位置上的坐标均为已知,具体坐标如下表:

表2 标准位置下无人机的位置坐标

编号	0	1	2	3	4
极坐标(m,°)	(0,0)	(100,0)	(100,40)	(100,80)	(100,120)

编号	5	6	7	8	9
极坐标(m,°)	(100,160)	(100,200)	(100,240)	(100,280)	(100,320)

若想将无人机调整至标准位置,则无人机需知道自己的坐标位置,但由于无人机接收到的只有角度信息,所以需对其进行定位,无人机才能获得自己所在位置的坐标,故第(3)小问的基础仍为定位问题。

根据第(1)小问可知,若想要定位一个存在位置偏差的点,至少需要3个无位置偏差的点,但分析题中所给的初始数据可知,只有编号为0和1的无人机无位置偏差,所以我们还需确定一个点作为无位置偏差点,即除编号为0,1的无人机外,还需从剩下的无人机中选出一架用于发射信号。

为了使最后的调整结果最大程度的逼近标准位置，我们在挑选剩下的无人机时，应挑选距离标准位置最近的点。所以我们通过计算无人机位置与标准位置之间的欧氏距离，选取欧式距离最小的无人机用于发射信号。

结合贪婪算法^[5]的思想，即可得到该问的调整方案，具体步骤如下：

Step1: 计算初始位置与标准位置之间的欧氏距离，选取欧式距离最小的无人机用于发射信号；

Step2: 根据编号 0，编号 1 和 Step1 中选取的三架无人机构建三角定位模型进行第一轮定位和调整；

Step3: 第一轮调整过后，将 Step1 中选取的无人机固定，计算剩下的无人机与标准位置的欧氏距离，仍然选择欧氏距离最小的无人机用于发射信号，与编号 0，编号 1 的无人机共同构建三角定位模型，从而进行第二轮的定位和调整；

以此类推，直至编号 2~8 的无人机均发射过 1 次信号，调整结束。

编写程序（具体程序见附录）对该问进行求解，得到如下的具体调整方案：

第一次调整：选定编号为 *FY02* 的无人机发射信号，对其余点进行定位和调整；

第二次调整：选定编号为 *FY05* 的无人机发射信号，对其余点进行定位和调整；

第三次调整：选定编号为 *FY06* 的无人机发射信号，对其余点进行定位和调整；

第四次调整：选定编号为 *FY08* 的无人机发射信号，对其余点进行定位和调整；

第五次调整：选定编号为 *FY07* 的无人机发射信号，对其余点进行定位和调整；

第六次调整：选定编号为 *FY03* 的无人机发射信号，对其余点进行定位和调整；

第七次调整：选定编号为 *FY04* 的无人机发射信号，对其余点进行定位和调整。

经过七次调整后，编号 2~8 的无人机均发射过 1 次信号，调整结束。调整过程及调整结果见下图（图中蓝色为标准位置，红色为无人机位置）：

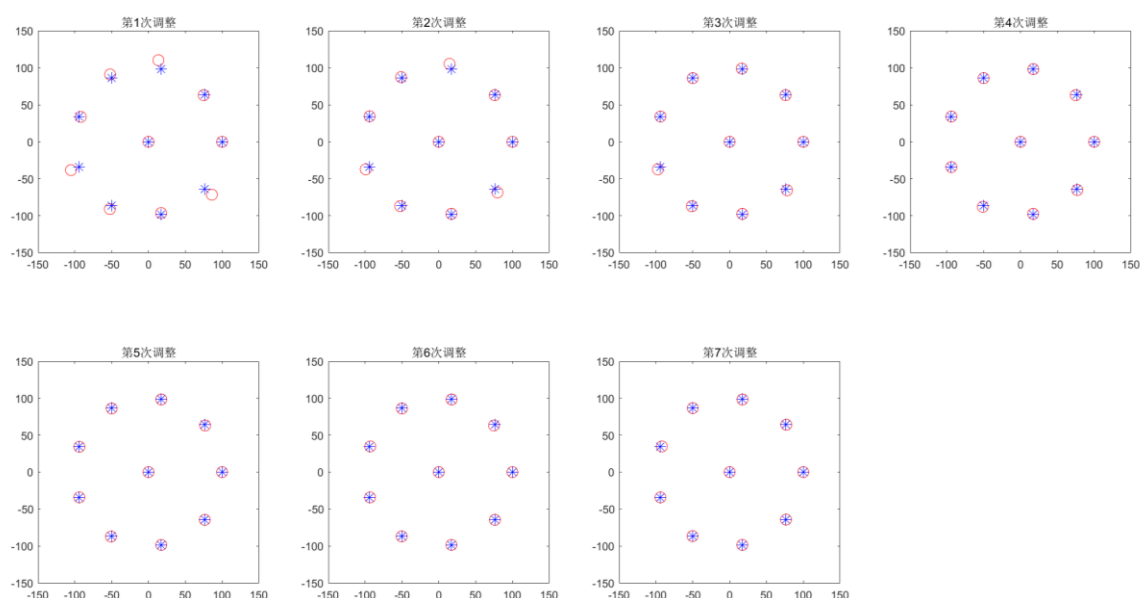


图 14 问题 1 第（3）问调整过程及结果

从图中可以看出，随着每增加 1 次调整，无人机的实际位置离标准位置就更近一步，且经过 7 次调整之后，无人机的实际位置与标准位置基本重合。

5.4 问题 2 模型的建立与求解

由问题 1 第（1）小问可知，要想定位一个存在位置偏差的点，至少需要三个位置

已知的点。故我们假设队列的顶点和与它相邻的两个点不存在位置偏差（即位置已知），即下图中的点 $FY1$ ，点 $FY2$ ，点 $FY3$ ，并以点 $FY1$ 为坐标原点， $FY2 - FY3$ 为 y 轴， $FY2 - FY3$ 的垂直平分线为 x 轴建立直角坐标系。则同问题 1 第（3）小问一样，无人机在标准位置下的坐标均为已知。

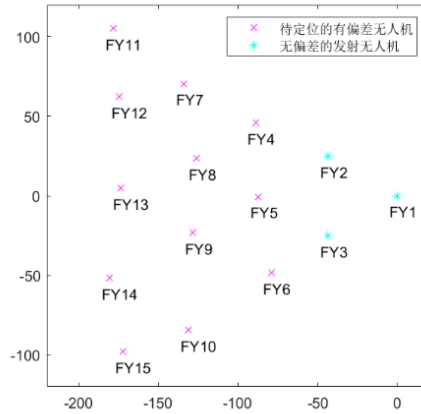


图 15 问题 2 待调整队列示意图

结合问题 1 第（1）小问建立的三角定位模型，即可给出该问的调整方案。

Step1: 根据 $FY1$ ， $FY2$ ， $FY3$ 可以确定 $FY5$ 的位置坐标，则可根据坐标将点 $FY5$ 调至标准位置。调整过后， $FY5$ 也作为已知点进行信号发射；

Step2: 选取点 $FY2$ ，点 $FY3$ 和点 $FY5$ 建立三角定位模型，可以求解出点 $FY4$ 和点 $FY6$ 的坐标，则可根据此将它们调整至标准位置。调整过后， $FY4$ ， $FY6$ 也作为已知点进行信号发射；

依此类推，直至所有无人机均经历了 1 次调整之后，调整结束，编写程序对问题进行求解（具体求解程序见附录），得到具体调整方案如下：

第一次调整：选取编号为 $FY1$ ， $FY2$ ， $FY3$ 的无人机发射信号，对编号为 $FY5$ 的无人机进行定位和调整；

第二次调整：选取编号为 $FY1$ ， $FY2$ ， $FY3$ ， $FY5$ 的无人机发射信号，对编号为 $FY4$ ， $FY6$ 的无人机进行定位和调整；

第三次调整：选取编号为 $FY1$ ， $FY2$ ， $FY3$ ， $FY4$ ， $FY5$ ， $FY6$ 的无人机发射信号，对编号为 $FY8$ ， $FY9$ 的无人机进行定位和调整；

第四次调整：选取编号为 $FY1$ ， $FY2$ ， $FY3$ ， $FY4$ ， $FY5$ ， $FY6$ ， $FY8$ ， $FY9$ 的无人机发射信号，对编号为 $FY7$ ， $FY10$ ， $FY13$ 的无人机进行定位和调整；

第五次调整：选取编号为 $FY1$ ， $FY2$ ， $FY3$ ， $FY4$ ， $FY5$ ， $FY6$ ， $FY8$ ， $FY9$ ， $FY10$ ， $FY13$ 的无人机发射信号，对编号为 $FY12$ ， $FY14$ 的无人机进行定位和调整；

第六次调整：选取编号为 $FY1$ ， $FY2$ ， $FY3$ ， $FY4$ ， $FY5$ ， $FY6$ ， $FY7$ ， $FY8$ ， $FY9$ ， $FY10$ ， $FY12$ ， $FY13$ ， $FY14$ 的无人机发射信号，对编号为 $FY11$ ， $FY15$ 的无人机进行定位和调整；

至此所有无人机均经历了 1 次调整，调整结束。由于该问没有给出具体的初始队列，故没有调整结果展示，我们将在模型的分析与检验中，对其采取计算机仿真，用于验证该调整方案的合理性。

六、模型的分析与检验

由于问题 1 第 (1)、(2) 小问的模型为公式推导，无法对其进行检验和分析，故我们只对问题 1 第 (3) 小问和问题 2 进行模型检验和分析。

6.1 问题 1 第 (3) 小问模型分析与检验

我们将误差定义为：

$$\text{误差} = \sum \text{调整后每个点实际位置与理想位置的欧氏距离} \quad (32)$$

计算每一次位置更新后的误差，得到如下图所示的误差曲线：

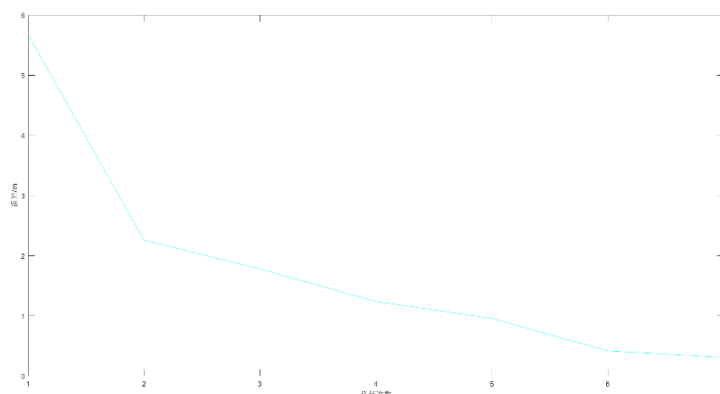


图 16 问题 1 第 (3) 小问误差曲线

从图中可以看出，误差随更新次数的增加逐渐减少，当经过 7 次位置更新之后，误差已经小于 0.5m，且 $\frac{0.5}{100} = 0.005 < 0.01$ ，即 0.5 相对于 100 是一个很小的数，所以经过 7 次位置更新之后，可认为调整后的实际位置与理想位置是重合的。证明了我们给出的调整方案具有可行性，且结果准确度高。

6.2 问题而模型分析与检验

为验证问题 2 给出的调整方案是否可行，我们对问题 2 建立的调整方案进行计算机仿真。

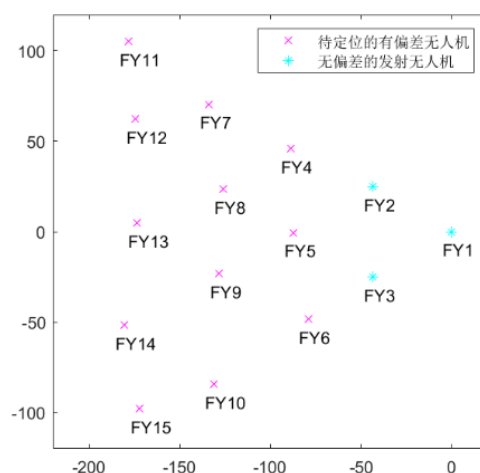


图 17 待调整的不标准锥形队列

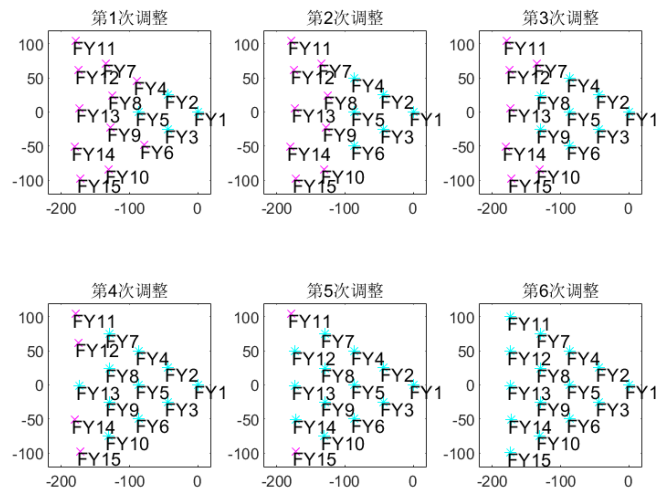


图 18 锥形队列调整过程

从图中我们可以看出，经过 6 次调整之后，形成的无人机已经近似为一个标准锥形编队，为了更加明确是否具有较好的调整效果，我们将无人机调整前的位置、调整后的位置以及理想位置在同一张图上表示，如图：

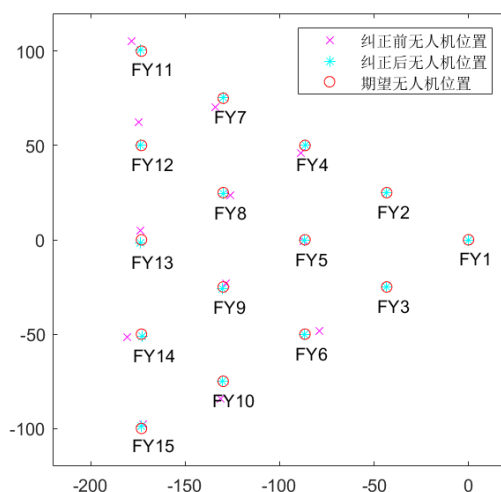


图 19 无人机调整前后效果对比图

从图中我们可以看出，最终的调整结果已经基本与理想位置重合，说明我们给出调整方案具有很高的可行性。

同时，我们仍然对该问进行误差分析，计算每次调整后的的误差，得到如下图所示的误差曲线：

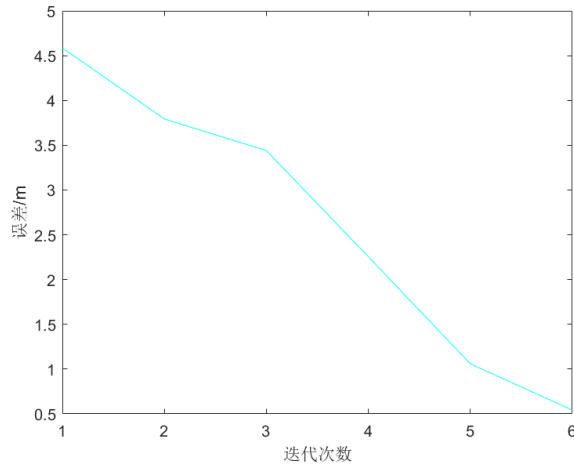


图 20 锥形队列调整误差曲线

从图中我们可以看出，经过 6 次调整后的误差已经近似为 0.5m,而 $\frac{0.5}{50} = 0.01$ ，即 0.5 相对于 50 来说是一个很小的数，所以可认为调整后的实际位置与理想位置是重合的。

综上，我们针对问题 2 给出调整方案具有可行性，且调整结果准确度高。

七、模型的评价与推广

7.1 模型的优点

(1) 问题 1 第 (1) 小问建立的三角定位模型原理简单，易于理解，且贯穿全文，使得全文连贯性高，逻辑性强；

(2) 建立的三角定位模型可以只通过方位角就实现无人机的有效定位；

(3) 问题 1 第 (3) 使用的贪婪算法易于实现且非常高效；

(4) 基于三角定位模型给出的队列调整方案可行性高，且调整结果准确度高。

7.2 模型的缺点

(1) 目标无源定位过程实质上为非线性最优估计问题，而最小二乘法主要采用线性近似法，若定位模型具有高强度的非线性会使得定位精度较低；

(2) 贪婪算法虽然简单高效，但给出的为接近最优的解决方案，并不是最优的解决方案。

7.3 模型的推广

本文建立的三角定位模型除了可以用于无人机定位，还可用于其他领域的定位（例如船舶定位）；同时，基于三角定位模型可以实现无人机位置调整，除了调整为问题 1 第 (3) 小问的圆形编队和问题 2 的锥形编队，我们还可以调整为三角形、椭圆，甚至特殊图形等，并将其应用于实际生活中（例如无人机表演）；最后，本文只有考虑二维平面内的定位问题，我们可以将我们建立的模型推广至三维，使得其应用范围更广。

八、参考文献

- [1]孙洪胜. 基于方位角测量的无源定位算法研究[D].哈尔滨工程大学,2010.
- [2]马晨.借助实验探究 再现“火热思考”——以“圆周角(第一课时)”为例[J].中小学数学(初中版),2022(06):47-49.
- [3]张鲲,沈重,王海丰,李壮,高倩,李涵雯.海上侦察船的纯方位无源定位技术研究[J].舰船科学技术,2018,40(02):19-21.
- [4]张卓然,叶广强,刘华伟.一种机载单站对固定目标的无源定位方法[J].计算机仿真,2015,32(02):92-95+105.
- [5]本刊综合.贪婪算法[J].发明与创新(高中生),2022(08):58.
- [6]Y. Oshman and P. Davidson, "Optimization of observer trajectories for bearings-only target localization," in *IEEE Transactions on Aerospace and Electronic Systems*, vol. 35, no. 3, pp. 892-902, July 1999, doi: 10.1109/7.784059.

附录

附录 1

介绍：支撑材料的文件列表

- 1、代码.rar：所有 matlab 的程序源代码，具体有六类：
 - (1) AngleCalu.m 为问题 2 计算待调整无人机与发射信号无人机的角度函数
 - (2) DisCal.m 用于计算待调整无人机与准确位置的欧氏距离
 - (3) init.m 为问题 1 第 (3) 小问无人机初始位置与准确位置对比图的文件
 - (4) locfun.m 为问题 2 计算待调整的无人机当前位置的函数
 - (5) main.m 为问题 2 的主函数，包括计算误差，仿真图以及调整距离
 - (6) wentil_3 用于计算问题 1 第 (3) 小问的最小误差
- 2、图.rar：文中出现的图。

附录 2

介绍：该代码由 Matlab 编写，用于求解问题 1 第 (3) 小问

```
% 无偏差的无人机
uav_loc = [
    0, 0, 0;
    2, 100, 0;
    3, 76.6044, 64.2787;
    4, 17.3648, 98.4807;
    5, -50, 86.6025;
    6, -93.9692, 34.2020;
    7, -93.9692, -34.2020;
    8, -50, -86.6025;
    9, 17.3648, -98.4807;
    10, 76.6044, -64.2787;
];

% 最开始有偏差的无人机
uav_loca = [
    1, 0, 0;
    2, 100, 0;
    3, 74.9602, 63.1218;
    4, 13.44, 110.3648;
    5, -52.101, 91.1505;
    6, -92.0024, 33.7414;
    7, -105.2688, -38.2256;
    8, -52.395, -90.993;
    9, 17.2088, -96.4516;
    10, 86.1392, -71.568;
];
```

```

for i = 3:10
    Err(i) = norm(uav_loca(i, 2:3) - uav_loc(i, 2:3));
end
fprintf('开始时的误差为%f\n', mean(Err));

angle = [0 0 40 80 120 160 -160 -120 -80 -40];
% 选出误差最小的无人机作为发射站
UavAdjust = [0 0 1 1 1 1 1 1 1 1];
MinErr = [1000 1000 0 0 0 0 0 0 0 0];
count = 1;
while(count <= 7)
    for i = 1:10
        if(UavAdjust(i) == 1)
            MinErr(i) = DisCal(uav_loc(i, 2:3), uav_loca(i, 2:3));
            MinError = min(MinErr);
        end
    end
    MinErrorInd = find(MinErr == MinError);
    fprintf('最小误差无人机编号为%d\n', MinErrorInd);

    UavAdjust(MinErrorInd) = 0;
    MinErr(MinErrorInd) = 1000;
    Index = find(UavAdjust == 1);
    UavFather = [
        0 0 0 0;
        0 0 0 0;
        1 2 MinErrorInd angle(MinErrorInd);
        1 2 MinErrorInd angle(MinErrorInd);
        1 2 MinErrorInd angle(MinErrorInd);
        1 2 MinErrorInd angle(MinErrorInd);
        1 2 MinErrorInd angle(MinErrorInd);
        1 2 MinErrorInd angle(MinErrorInd);
        1 2 MinErrorInd angle(MinErrorInd);
        1 2 MinErrorInd angle(MinErrorInd);
    ];
    for i = Index
        Angle = AngleCalu(uav_loca(i, 2:3), UavFather(i, 1:3), uav_loca(:, 2:3));
        Loc = locfun(Angle, UavFather(i, 1:3), uav_loc(:, 2:3), UavFather(i, 4));
        MovDis = uav_loc(i, 2:3) - Loc;
        uav_loca(i, 2:3) = uav_loca(i, 2:3) + MovDis;
    end

    for i = 3:10
        dataseq(count).Err(i) = norm(uav_loca(i, 2:3) - uav_loc(i, 2:3));
    end
end

```

```

end

fprintf('调整后的误差为%f\n', mean(dataseq(count).Err));
fprintf('-----\n');
count = count + 1;
end

```

附录 3

介绍：该代码由 Matlab 编写，用于绘制问题 1 第（3）小问的效果对比图

```

function drone = init()
theta = 0:360 / 9:320;
rho = 100;
drone.FY00.rho = 0;
drone.FY01.rho = 100;
drone.FY02.rho = 100;
drone.FY03.rho = 100;
drone.FY04.rho = 100;
drone.FY05.rho = 100;
drone.FY06.rho = 100;
drone.FY07.rho = 100;
drone.FY08.rho = 100;
drone.FY09.rho = 100;
drone.FY00.theta = 0;
drone.FY01.theta = theta(1);
drone.FY02.theta = theta(2);
drone.FY03.theta = theta(3);
drone.FY04.theta = theta(4);
drone.FY05.theta = theta(5);
drone.FY06.theta = theta(6);
drone.FY07.theta = theta(7);
drone.FY08.theta = theta(8);
drone.FY09.theta = theta(9);

% 标准点的坐标
drone.FY00.x = drone.FY00.rho*cosd(drone.FY00.theta);
drone.FY01.x = drone.FY01.rho*cosd(drone.FY01.theta);
drone.FY02.x = drone.FY02.rho*cosd(drone.FY02.theta);
drone.FY03.x = drone.FY03.rho*cosd(drone.FY03.theta);
drone.FY04.x = drone.FY04.rho*cosd(drone.FY04.theta);
drone.FY05.x = drone.FY05.rho*cosd(drone.FY05.theta);
drone.FY06.x = drone.FY06.rho*cosd(drone.FY06.theta);
drone.FY07.x = drone.FY07.rho*cosd(drone.FY07.theta);
drone.FY08.x = drone.FY08.rho*cosd(drone.FY08.theta);

```



```

drone.FY09.x = drone.FY09.rho*cosd(drone.FY09.theta);
drone.FY00.y = drone.FY00.rho*sind(drone.FY00.theta);
drone.FY01.y = drone.FY01.rho*sind(drone.FY01.theta);
drone.FY02.y = drone.FY02.rho*sind(drone.FY02.theta);
drone.FY03.y = drone.FY03.rho*sind(drone.FY03.theta);
drone.FY04.y = drone.FY04.rho*sind(drone.FY04.theta);
drone.FY05.y = drone.FY05.rho*sind(drone.FY05.theta);
drone.FY06.y = drone.FY06.rho*sind(drone.FY06.theta);
drone.FY07.y = drone.FY07.rho*sind(drone.FY07.theta);
drone.FY08.y = drone.FY08.rho*sind(drone.FY08.theta);
drone.FY09.y = drone.FY09.rho*sind(drone.FY09.theta);

% 有误差点的坐标
drone.FY0.x = 0; drone.FY0.y = 0;
drone.FY1.x = 100; drone.FY1.y = 0;
drone.FY2.x = 74.9602; drone.FY2.y = 63.1218;
drone.FY3.x = 13.44; drone.FY3.y = 110.3648;
drone.FY4.x = -52.101; drone.FY4.y = 91.1505;
drone.FY5.x = -92.0024; drone.FY5.y = 33.7414;
drone.FY6.x = -105.2688; drone.FY6.y = -38.2256;
drone.FY7.x = -52.395; drone.FY7.y = -90.993;
drone.FY8.x = 17.2088; drone.FY8.y = -96.4516;
drone.FY9.x = 86.1392; drone.FY9.y = -71.568;

figure(1)
axis equal
hold on
axis square;
axis([-150 150 -150 150])
drone.FY00.h = plot(drone.FY00.x,drone.FY00.y,'ro','MarkerSize',10);
drone.FY00.text = text(drone.FY00.x,drone.FY00.y,'FY00');
drone.FY0.h = plot(drone.FY0.x, drone.FY0.y, 'c*', 'MarkerSize', 10);
drone.FY0.text = text(drone.FY0.x, drone.FY0.y, 'FY0');

drone.FY01.h = plot(drone.FY01.x,drone.FY01.y,'ro','MarkerSize',10);
drone.FY01.text = text(drone.FY01.x,drone.FY01.y,'FY01');
drone.FY1.h = plot(drone.FY1.x, drone.FY1.y, 'c*', 'MarkerSize', 10);
drone.FY1.text = text(drone.FY1.x, drone.FY1.y, 'FY1');

drone.FY02.h = plot(drone.FY02.x,drone.FY02.y,'ro','MarkerSize',10);
drone.FY02.text = text(drone.FY02.x,drone.FY02.y,'FY02');
drone.FY2.h = plot(drone.FY2.x, drone.FY2.y, 'c*', 'MarkerSize', 10);
drone.FY2.text = text(drone.FY2.x, drone.FY2.y, 'FY2');

```

```

drone.FY03.h = plot(drone.FY03.x,drone.FY03.y,'ro','MarkerSize',10);
drone.FY03.text = text(drone.FY03.x,drone.FY03.y,'FY03');
drone.FY3.h = plot(drone.FY3.x, drone.FY3.y, 'c*', 'MarkerSize', 10);
drone.FY3.text = text(drone.FY3.x, drone.FY3.y, 'FY3');

drone.FY04.h = plot(drone.FY04.x,drone.FY04.y,'ro','MarkerSize',10);
drone.FY04.text = text(drone.FY04.x,drone.FY04.y,'FY04');
drone.FY4.h = plot(drone.FY4.x, drone.FY4.y, 'c*', 'MarkerSize', 10);
drone.FY4.text = text(drone.FY4.x, drone.FY4.y, 'FY4');

drone.FY05.h = plot(drone.FY05.x,drone.FY05.y,'ro','MarkerSize',10);
drone.FY05.text = text(drone.FY05.x,drone.FY05.y,'FY05');
drone.FY5.h = plot(drone.FY5.x, drone.FY5.y, 'c*', 'MarkerSize', 10);
drone.FY5.text = text(drone.FY5.x, drone.FY5.y, 'FY5');

drone.FY06.h = plot(drone.FY06.x,drone.FY06.y,'ro','MarkerSize',10);
drone.FY06.text = text(drone.FY06.x,drone.FY06.y,'FY06');
drone.FY6.h = plot(drone.FY6.x, drone.FY6.y, 'c*', 'MarkerSize', 10);
drone.FY6.text = text(drone.FY6.x, drone.FY6.y, 'FY6');

drone.FY07.h = plot(drone.FY07.x,drone.FY07.y,'ro','MarkerSize',10);
drone.FY07.text = text(drone.FY07.x,drone.FY07.y,'FY07');
drone.FY7.h = plot(drone.FY7.x, drone.FY7.y, 'c*', 'MarkerSize', 10);
drone.FY7.text = text(drone.FY7.x, drone.FY7.y, 'FY7');

drone.FY08.h = plot(drone.FY08.x,drone.FY08.y,'ro','MarkerSize',10);
drone.FY08.text = text(drone.FY08.x,drone.FY08.y,'FY08');
drone.FY8.h = plot(drone.FY8.x, drone.FY8.y, 'c*', 'MarkerSize', 10);
drone.FY8.text = text(drone.FY8.x, drone.FY8.y, 'FY8');

drone.FY09.h = plot(drone.FY09.x,drone.FY09.y,'ro','MarkerSize',10);
drone.FY09.text = text(drone.FY09.x,drone.FY09.y,'FY09');
drone.FY9.h = plot(drone.FY9.x, drone.FY9.y, 'c*', 'MarkerSize', 10);
drone.FY9.text = text(drone.FY9.x, drone.FY9.y, 'FY9');

plot(rho*cosd(0:10:360),rho*sind(0:10:360));
savefig('3_first.fig');
end

```

附录 4

介绍：该代码由 Matlab 编写，用于求解问题 2

```

clear all;
clc;

```

```

close all;

%% 设定参数
% 期望位置
R = 50; % 距离为 50
R0 = R / 2;
R1 = R * sqrt(3) / 2;

uav_loc = [
    1 0 0;
    2 -R1 R0;
    3,-R1, -R0;
    4,-R1 * 2, R0 * 2;
    5,-R1 * 2, 0;
    6,-R1 * 2, -R0 * 2;
    7,-R1 * 3, R0 * 3;
    8,-R1 * 3, R0 * 1;
    9,-R1 * 3, -R0 * 1;
    10,-R1 * 3, -R0 * 3;
    11,-R1 * 4, R0 * 4;
    12,-R1 * 4, R0 * 2;
    13,-R1 * 4, R0 * 0;
    14,-R1 * 4, -R0 * 2;
    15,-R1 * 4, -R0 * 4;
];

% 第一次迭代发射信号的三架无人机
leadInd = [1 2 3];
figure(1);
clf;
hold on;
box on;
plot(uav_loc(4:end, 2), uav_loc(4:end, 3), 'mx');
plot(uav_loc(leadInd, 2), uav_loc(leadInd, 3), 'c*');
for ii=1:15
    text(uav_loc(ii, 2)-5, uav_loc(ii, 3)-10, ['FY' num2str(ii)]);
end
axis square;
axis([-220 20 -120 120])
legend('待定位的无人机','发射无人机')
title('无人机场景图');
% 保存待定位无人机与发射信号无人机的位置（此时都是准确位置）
savefig('1-1.fig');

```

```

%% 生成有偏差的数据
noisesigma = 5;
uav_loc = uav_loc;
% 生成随机的偏差位置
uav_loc(4:end, 2:3) = uav_loc(4:end, 2:3) + randn(12,2) * noisesigma;
figure(2);
clf;
hold on;
box on;
plot(uav_loc(4:end, 2), uav_loc(4:end, 3), 'mx');
plot(uav_loc(leadInd, 2), uav_loc(leadInd, 3), 'c*');
for ii = 1:15
    text(uav_loc(ii,2) - 5, uav_loc(ii,3) - 10, ['FY' num2str(ii)]);
end
axis square;
axis([-220 20 -120 120])
legend('待定位的有偏差无人机','无偏差的发射无人机')
savefig('1-2.fig');
uav_locr2 = uav_loc;

%% 在一个菱形中，需要调整的无人机接收另外三个点发射信号的无人机的编号
UavFather = [
    0 0 0 0;
    0 0 0 0;
    0 0 0 0;
    3 2 5 60;
    1 2 3 0;
    2 5 3 -60;
    5 4 8 60;
    2 4 5 0;
    3 5 6 0;
    5 9 6 -60;
    8 7 12 60;
    4 7 8 0;
    5 8 9 0;
    6 9 10 0;
    9 14 10 -60;];

% 最开始的情况，前三个无人机是发射信号的，设为 0
UavAdjust = [0 0 0 1 1 1 1 1 1 1 1 1 1 1];
figure(3);
clf;
count = 1;

```

```

% 迭代条件
while(sum(UavAdjust) > 0)
    % 找到矩阵中非零元素的索引
    Index = find(UavAdjust == 1);
    UavAdjustTemp = UavAdjust;
    fprintf('第%d 次调整,发射无人机['',count);
    % 发射信号的无人机
    for ii = 1:15
        if (UavAdjust(ii) == 0)
            fprintf('FY%d,',ii);
        end
    end
    fprintf('\b]\n 接收无人机['');

    AdjustIndex = [];
    for ii = Index
        if(sum(UavAdjust(UavFather(ii,1:3))) == 0)
            AngD = AngleCalu(uav_loc(ii,2:3), UavFather(ii,1:3), uav_loc(:,2:3));
            % 计算无人机当前所在的位置
            Loc=locfun(AngD, UavFather(ii,1:3), uav_loc(:,2:3), UavFather(ii,4));
            % 需要调整的距离, 即准确位置减去当前位置的距离之差
            MovDz = uav_loc(ii,2:3) - Loc;
            % 移动接收信号的无人机, 调整位置
            uav_loc(ii,2:3) = uav_loc(ii,2:3) + MovDz;
            % 将接收过信号的无人机设为 0
            UavAdjustTemp(ii) = 0;
            AdjustIndex = [AdjustIndex ii];
        end
    end
    UavAdjust = UavAdjustTemp;
    for ii = 1:15
        % 对移动后的无人机位置与准确位置的向量进行取模运算, 计算误差
        dataseq(count).Err(ii) = norm(uav_loc(ii,2:3) - uav_loc(ii,2:3));
    end

    for ii = AdjustIndex
        fprintf('FY%d,',ii);
    end
    % 对当前误差去平均值
    fprintf('\b]\n 调整后误差%fm\n', mean(dataseq(count).Err));

    subplot(2, 3, count);
    hold on;

```

```

    box on;
    plot(uav_loca(UavAdjust == 1,2), uav_loca(UavAdjust == 1,3), 'mx');
    plot(uav_loca(UavAdjust == 0,2), uav_loca(UavAdjust == 0,3), 'c*');
    for ii = 1:15
        text(uav_loca(ii,2) - 5, uav_loca(ii,3) - 10, ['FY' num2str(ii)]);
    end
    axis square;
    axis([-220 20 -120 120])
    title(['第' num2str(count) '次调整']);
    % 每次调整完次数加一
    count = count + 1;
end
savefig('1-3.fig');

% 对误差的结果统计分析
for it = 1:count - 1
    res(it) = mean(datasseq(it).Err);
end
figure(4);
plot(res, 'c');
xlabel('迭代次数');
ylabel('误差/m');
savefig('1-4.fig');

figure(5);
clf;
hold on;
box on;
plot(uav_locr2(:,2), uav_locr2(:,3), 'mx');
plot(uav_loca(:,2), uav_loca(:,3), 'c*');
plot(uav_loc(:,2), uav_loc(:,3), 'ro');

for ii = 1:15
    text(uav_loca(ii,2) - 5, uav_loca(ii,3)-10, ['FY' num2str(ii)]);
end
axis square;
axis([-220 20 -120 120])
legend('纠正前无人机位置','纠正后无人机位置','期望无人机位置')
savefig('1-5.fig');

```