

# 基于信息源回溯与迭代寻优的无人机无源定位及编队调整问题

Wenjie Lan; Zhixi Li; Xudong Yin

## 摘 要

本文建模型、创新算法解决了无人机遂行编队飞行中无源定位与编队调整问题。

针对问题一第（1）问，经分析该问可以转化为多个弧轨迹求交点的问题。基于此，我们在极坐标系下建立了具有普适性的**多点轨迹定位模型**，并模型建立过程中创新性地开发了**信息源回溯算法**（附录 2），该算法通过将理想位置下各无人机组组合发射信号构成的角度与被定为无人机实际获得的角度在一定误差范围内的比较，匹配到各角度对应的发射源无人机编号，从而求解出对应的轨迹方程。最终轨迹方程联立，得到**最终的多点轨迹定位模型**（5.1.2）实现对无人机的定位。

针对问题一第（2）问，我们将问题转化为了信号源无人机最优增量规划问题，建立**0-1规划模型**（5.2.1），并利用**循环搜索结构**进行求解。过程中，我们结合本问的条件改进了**信息源回溯算法**（附录3）：在无人机位置略有偏差的情况下，对未知编号的无人机通过方位角的近似分析确定其编号。在此基础上，求解规划模型得到除FY00和 FY01外，仅增加1架无人机发射信号，就能实现有效定位（5.2.3）。

针对问题一第（3）小问，我们基于贪婪思想建立了无人机的**定位迭代寻优模型**（5.3.1）。首先，固定选择位置无偏的 FY00 与 FY01 确定坐标系并作为固定的信号发射源；接着，选择另一偏差最小的 FY02 作第三个信号源，对无人机进行近似定位；定位后，对其位置进行调整，此为一次迭代。接下来重复步骤，先后选择 FY06、FY03 作第三个信号源，共完成三次迭代，误差收敛，实现了队形有效调整，该迭代过程就是我们提出的调整方案（5.3.2）。

针对问题二，我们基于之前的信息源回溯算法与定位模型，得到了锥形编队情况下的无人机调整方案（5.4.1）：首先通过构成最中心或次中心等边三角形的三架无人机进行定位，再依次向外一层进行调整，直到达到最外层停止。接着，我们将已构建算法与模型进行推广的任意队形（5.4.2），第一步建立初始数量大于四架无人机群的位置迭代算法，第二步在已在准确位置中的无人机中选择三架无人机，对其余相邻且未进行位置调整的无人机进行**精确位置调整**，重复第二步至所有无人机均有效地调整位置。最后，我们模型进一步推广至现实中的编队应用，举例了“中国”型编队的无人机调整问题（5.4.3）。

最后，我们对模型进行稳健性检验。通过对误差角度的讨论验证了我们对问题一的第（2）问的改进信息源回溯算法在无人机略有偏差情况下的可靠性与稳健性。

全文以信息源回溯算法与多动点轨迹定位模型为核心，结合了问题一中的实际情景，优化了无人机编队的调整方案。同时将定位模型与迭代算法在调整方案中的应用进行有效的推广。本文的模型与求解对现实无人机定位与编队问题具有一定意义。

关键词：信息源回溯算法 多点轨迹定位模型 0-1 规划 定位迭代寻优模型

## 一、问题重述

### 1.1 背景资料

无源定位是一种用无线电接收设备，接收对方电子设备发射的电磁波来确定其位置的定位方法。在无人机的编队飞行中，为了避免外界的干扰，应尽可能少向外发射电磁波信号，因此，纯方位无源定位的方法在保持无人机编队队形的过程中发挥重要作用。

### 1.2 需要解决的问题

根据题目信息，问题总结如下：

**问题一：**已知十架无人机进行圆形编队，其中一架位于圆心，其余九架均匀分布在圆上，求解：

- (1) 位于圆心的无人机（FY00）与编队中另外两架编号已知、位置无偏的无人机向位置有偏的无人机发送信号，建立模型以定位接被动收信号的无人机。
- (2) 某位置有偏的无人机接收到编号 FY00、FY01 与另外若干编号未知、位置无偏的无人机信号，建立模型求解除了 FY00 与 FY01 之外还需要几架无人机以实现有效定位。
- (3) 根据无人机的初始位置，按照编队要求，通过多次调整，每次选择编号为 FY00 的无人机和圆周上不超过 3 架无人机发射信号，其余无人机根据接收到的方向信息，调整到理想位置（忽略调整时间），使得 9 架无人机最终均匀分布在某个圆周上。

**问题二：**已知无人机进行锥形编队，直线上相邻两架无人机等间距间隔，求解：仍使用无源定位的方法，建立模型给出无人机位置调整方案。

## 二、模型假设

1. 每一架无人机都视为一个质点，即忽略其本身的体积。
2. 十架无人机均在相同的高度、以相同的速度、相同的方向做相同性质的运动，即十架无人机构成一个相对静止的二维运动系统。
3. 位置有偏的无人机仅是略有偏差，根据测量学相关理论，假设题目中的“略有偏差”的最大误差不超过 $5^\circ$ 。

## 三、符号说明

符号	说明	单位
$H$	需要定位的、被动接收信号的无人机	-
$L_1$	包含所有位置无偏的无人机与 $H$ 所产生的夹角信息的集合	-
$T_{ij}^{(k)}$	第 $k$ 组 FY0 <i>i</i> 和 FY0 <i>j</i> 组合传递给 $H$ 的夹角信息	-
$\varphi$	包含所有轨迹方程能够组合成的方程组的集合	-
$T_{abc}^{(k)}$	$a, b, c$ 三架无人机依次作顶点所形成的角度信息的集合	-

## 四、 问题分析

### 2.1 问题一第（1）问的分析

针对第（1）问，我们建立多点轨迹定位模型、设计了信息源回溯算法。

我们先假设十架无人机均在相同的高度、以相同的速度、相同的方向做相同性质的运动，即十架无人机构成一个相对静止的二维运动系统。在得到角度与编号信息后，被动接收信息的无人机（即角的顶点）的轨迹方程可以被求解。

在三个信号源向被动接收信号的无人机发送信号时，无人机获得三个角度信息，从而产生三个轨迹方程，而三个轨迹方程的交点即该无人机的定位。

### 2.2 问题一第（2）问的分析

第（2）问要我们求解增加多少无人机能实现有效定位。经分析，增加的信息源无人机应该尽可能少。对于此，我们建立一个 0-1 规划模型，并设计出循环算法求解。

在每一次的循环中，都涉及到求解轨迹方程并联合进行定位的过程，这既需要角度信息，又需要角度对应的两架信息源无人机编号（同第（1）问中的建模分析）。

但第（2）问的条件中，被动接收信号的无人机不知道发送信号的信息源无人机的编号，因此我们改进第（1）问中设计的信息源回溯算法，在匹配角度与编号信息之前先通过误差范围内的比较确定编号。并在此基础上探索增加多少架无人机能够实现有效定位。

### 2.3 问题一第（3）问的分析

第（3）问条件中，除了FY00与FY01初始位置正确以外，其他八架无人机位置均有偏。故第（1）问所建立的定位模型无法应用于此，因为一个角度所产生的轨迹方程无法定位（因为仅有一个轨迹方程，无法产生交点）。

因此，我们需要选择略有偏差的无人机发射信号。首先选取原始位置与理想无偏位置偏差最小的无人机（即FY02）作为除FY00与FY01外的第三个信息源，与FY00、FY01一起向其他无人机发送信号，通过定位模型与正弦定理等方法建立方程组，确定其直角坐标系下的位置，即进行第一次迭代，迭代完成后选取误差最小的点作为新的信息源进行下一次迭代。

如此往复，在多次迭代后，最终各无人机的调整位置无限收敛于理想无偏位置，即此时每一架无人机实际位置距自己的理想位置的偏差都足够小。

### 2.4 问题二的分析

分析题目所给的锥形无人机编队可知，我们仍然可以采用问题一第（1）问所建立的定位模型与第（3）问的迭代算法进行无人机位置调整，从内往外逐层调整，而后可以总结到最基础的四边形无人机编队，发散到更多其他能够应用于实际场景的无人机编队，总结调整方案。

## 五、模型的建立与求解

### 5.1 问题一第（1）问：多点轨迹定位模型的建立

被动接收信号的无人机所接收到的信息仅为角度信息，由解析几何的知识，一个二维平面中，已知三点一角，我们可以求得该角顶点的轨迹方程[1]。

问题一第（1）问中，我们假设十架无人机均在相同的高度、以相同的速度、相同的方向做相同性质的运动，即十架无人机构成一个相对静止的二维运动系统。故被动接收信息的无人机（即角的顶点）的轨迹方程可以被求解。

在三个信号源向被动接收信号的无人机发送信号时，无人机获得三个角度信息，在此角度下，产生三个轨迹方程，而三个轨迹方程的交点即该无人机的定位。

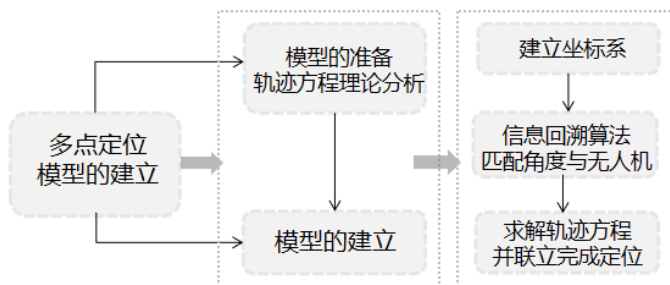


图 1. 问题一第（1）问流程图

#### 5.1.1 模型的准备

根据对问题的分析，本题可以转化为求动点的弧轨迹方程的交点，以下我们先从理论上对弧轨迹方程的求解情况进行讨论（均在极坐标系中讨论）。

首先需要考虑的是轨迹方程的个数，因为这会对我们最后的交点情况，即具体的定位情况产生影响。我们讨论如下：

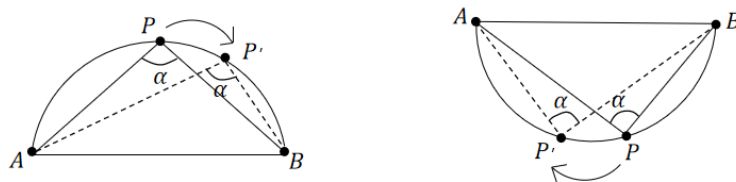


图 2. 弧轨迹方程的个数情况

情况一：动点位于弦 $AB$ 上方（图 1 左图）

当动点 $P(\rho, \theta)$ 位于弦 $AB$ 上方时，其轨迹构成一段圆弧，该弧轨迹方程可以按如下步骤求解：首先我们建立坐标系。此处为了便于展示如何推导动点 $P$ 的轨迹极坐标方程，我们不妨以点 $B$ 为极点建立极坐标系（正式建模时同理，只是极轴发生旋转）。

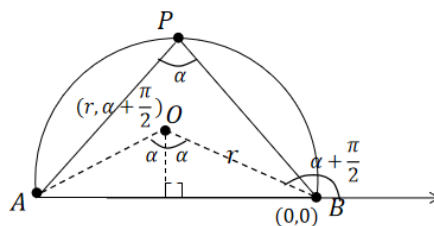


图 3. 弧轨迹方程的求解

为了最终的轨迹方程中包含角度 $\alpha$ ，我们利用圆周角定理（即 $\angle AOB = 2\angle APB$ ）建立轨迹方程与 $\alpha$ 的联系。紧接着，做圆心 $O$ 垂直于弦 $AB$ ，解析三角形后（如图 2），我们可得圆心的极坐标为： $(r, \frac{\pi}{2} + \alpha)$ ，直角坐标为 $(-r \cdot \cos\alpha, r \cdot \sin\alpha)$ ；

将坐标带入以下方程求解：

$$\begin{aligned} (x + r \cdot \cos\alpha)^2 + (y - r \cdot \sin\alpha)^2 &= r^2 \\ x &= \rho \cdot \cos\alpha, y = \rho \cdot \sin\alpha \end{aligned} \quad (1)$$

最终求得弧的轨迹方程为：

$$\rho = 2r \cdot \cos(\theta + \alpha) \quad (2)$$

**情况二：动点位于弦 $AB$ 下方（图 1 右图）**

当动点 $P(\rho, \theta)$ 位于弦 $AB$ 下方时，其轨迹同样构成一段弧，因此与情况一同理，可以求该弧的圆心为 $(r, -\frac{\pi}{2} - \alpha)$ ，并求得其轨迹方程为：

$$\rho_1 = 2r \cdot \cos(\theta - \alpha) \quad (3)$$

综上可知，动点位于弦的上、下方时，将有两个不同的轨迹方程。

接着是圆弧的形态（即优劣情况），这也会影响到我们交点的确定。因此我们对于优劣弧进行讨论：

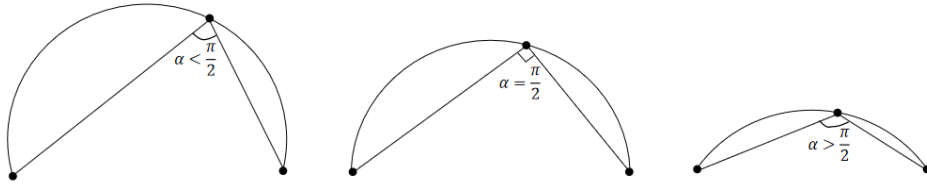


图 4. 优劣弧情况

由图像清楚可见：

- (1) 当 $0 \leq \alpha < \frac{\pi}{2}$ 时，动点轨迹为优弧；
- (2) 当 $\alpha = \frac{\pi}{2}$ 时，动点轨迹为半圆；
- (3) 当 $\frac{\pi}{2} < \alpha \leq \pi$ 时，动点轨迹为劣弧。

#### 5.1.2 模型的建立

**Step1：建立极坐标系：**

首先，我们设除了 FY00 发射信号以外，另外两架发送信号的无人机为：

$$\begin{aligned} &FY0K_1 ; FY0K_2 \\ &K_1, K_2 \in \{1, 2, \dots, 9\} \text{ 且 } K_1 \neq K_2 \end{aligned} \quad (4)$$

接着，我们以 FY00 为极点建立极坐标系：

因为 $FY0K_1$ 、 $FY0K_2$ 位置无偏，即均匀分布在圆形编队上，所以其极径均为圆形编队的半径 $r$ （ $r = 100$  米）；进一步地，设 $FY0K_1$ 、 $FY0K_2$ 的极角分别为 $\theta_{K_1}$ 和 $\theta_{K_2}$ 。由此，我们可以得到 $FY0K_1$ 、 $FY0K_2$ 的极坐标分别为： $(r, \theta_{K_1})$ 、 $(r, \theta_{K_2})$

**Step2：编写信息源回溯算法匹配角度信息与无人机编号**

我们用 $H$ 表示位置有偏、被动接收信号的无人机。据题目条件， $H$ 接收到的仅有

信号源的夹角信息，而没有编号信息，但信息源的事先编号已知，即 $H$ 事先已知哪三架无人机会向自己发送信号，但收到的信息仅有三个夹角大小的信息，而不知道这三个夹角分别是由三架无人机中哪两架无人机组组合发出的。

而在求轨迹方程时，除了需要夹角信息以外，我们还需要知道该夹角对应的两个信号源（即两个信号源无人机构成的弦的信息）。因此，为了后续求解轨迹方程，我们编写信息源回溯算法（**Matlab 代码见附录 2**）对角度信息与发送信号的无人机编号进行匹配。该算法流程如下：

**第一步**，建立集合 $L_1$ ，储存所有位置无偏的无人机与需要定位的无人机 $H$ 所产生的夹角信息：

$$L_1 = T_{ij}^{(1)} \cup T_{ij}^{(2)} \cup T_{ij}^{(3)} \cup \dots \cup T_{ij}^{(n)} \quad (5)$$

$$(i, j = 0, 1, 2, \dots, 9 \text{ 且 } i, j \neq H)$$

其中， $T_{ij}^{(k)}$ 表示第 $k$ 组 $FY0i$ 和 $FY0j$ 组合传递给 $H$ 的夹角信息。例如， $T_{01}^{(1)}$ 即表示第1组 $FY00$ 和 $FY01$ 向无人机 $H$ 发送信号所得到的夹角信息。

**第二步**，根据 $H$ 所获取的夹角信息 $\{\alpha_1, \alpha_2, \alpha_3\}$ ，给定误差 $\varepsilon$ 并依次在 $L_1$ 中进行误差范围内的匹配，即判断以下关系是否成立：

$$\alpha_m + \varepsilon \in L_1 \quad (m = 1, 2, 3) \quad (6)$$

**第三步**，若上述关系成立，则挑选出 $L_1$ 中与 $\alpha_m$ 匹配的角度所对应的两架无人机；若上述关系不成立，则增大误差项并进入第四步：

$$\varepsilon = \varepsilon + \Delta \quad (7)$$

**第四步**，回到第二步，比较加入新的误差项后(7)式关系是否成立，直到所有的夹角信息 $\{\alpha_1, \alpha_2, \alpha_3\}$ 全部匹配成功时截止。

流程可可视化如下图：

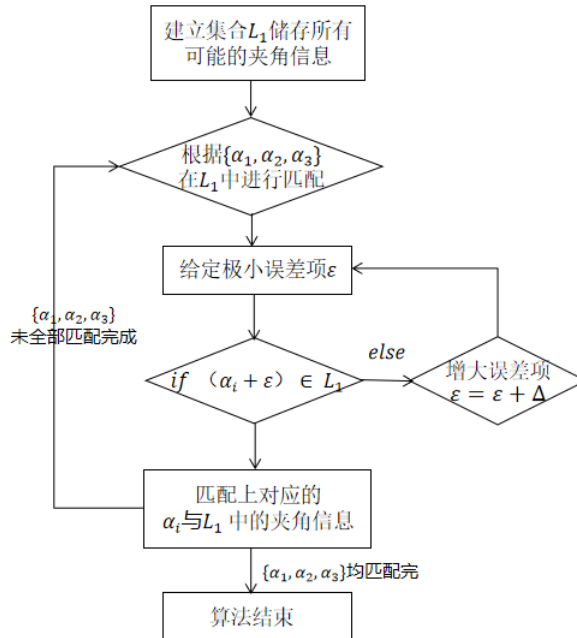


图 5. 信息源回溯算法

### Step3: 求解弧轨迹方程

在 Step2 中的算法完成后， $H$ 所有的角度信息完成其信号源的匹配。以下过程中我们不妨设其所接收到 $FY0K_1$ 与 $FY00$ 的信号夹角为 $\alpha_1$ ；接收到 $FY0K_2$ 与 $FY00$ 的信

号的夹角为 $\alpha_2$ ；接收到 $FY0K_1$ 与 $FY0K_2$ 的信号夹角为 $\alpha_3$ 。设被动接收信号的无人机 $H$ 的极坐标为 $(\rho, \theta)$ 。

由 5.1.1 的理论分析，我们可以得到被动接收信号的无人机 $H$ 对应的三条弧的轨迹方程为：

夹角 $\alpha_1$ （即被动接收信号的 $H$ 与 $FY0K_1$ 、 $FY00$ 所形成的夹角）所形成的第一个弧的轨迹方程（当 $H$ 在弦 $FY0K_1$ - $FY00$ 的上方时）为：

$$\rho_1 = r \cdot \sin \alpha_1 \cdot \cos(\theta - \theta_{K_1} - \frac{\pi}{2} + \alpha_1) \quad (8)$$

以及第二个弧的轨迹方程（当 $H$ 在弦 $FY0K_1$ - $FY00$ 的下方时）：

$$\rho_1 = r \cdot \sin \alpha_1 \cdot \cos(\theta - \theta_{K_1} - \frac{\pi}{2} + \alpha_1) \quad (9)$$

其中，

$$\begin{cases} \rho_1 \text{为优弧} & 0 \leq \alpha_1 < \frac{\pi}{2} \\ \rho_1 \text{为半圆} & \alpha_1 = \frac{\pi}{2} \\ \rho_1 \text{为劣弧} & \frac{\pi}{2} < \alpha_1 \leq \pi \end{cases} \quad (10)$$

夹角 $\alpha_2$ （即被动接收信号的 $H$ 与 $FY0K_2$ 、 $FY00$ 所形成的夹角）所形成的第一个弧的轨迹方程（当 $H$ 在弦 $FY0K_2$ - $FY00$ 的上方时）为：

$$\rho_2 = r \cdot \sin \alpha_2 \cdot \cos(\theta - \theta_{K_2} + \frac{\pi}{2} - \alpha_2) \quad (11)$$

以及第二个弧的轨迹方程（当 $H$ 在弦 $FY0K_2$ - $FY00$ 的下方时）：

$$\rho_2 = r \cdot \sin \alpha_2 \cdot \cos(\theta - \theta_{K_2} - \frac{\pi}{2} + \alpha_2) \quad (12)$$

其中， $\rho_2$ 的优劣弧情况与 $\rho_1$ 相同。

$\alpha_1$ 、 $\alpha_2$ 产生的轨迹方程示意图如下：

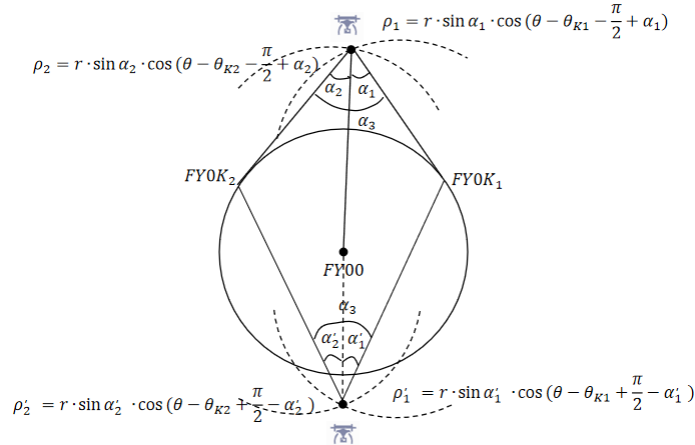


图 6. 轨迹方程示意图

图中的虚线为即 $H$ 与相应的信号源无人机所形成的弧的轨迹方程。

夹角 $\alpha_3$ （即被动接收信号的 $H$ 与 $FY0K_1$ 、 $FY0K_2$ 所形成的夹角）所形成的第一个弧的轨迹方程（当 $H$ 在弦 $FY0K_1$ - $FY0K_2$ 的上方时）：

$$\rho_3 = (1 + \frac{\tan |\frac{\theta_{K_1} - \theta_{K_2}}{2}|}{\tan \alpha_3}) \cdot r \cdot \sqrt{2 + \cos(\theta_{K_1} - \theta_{K_2})} \cdot \cos(\theta - \arctan \frac{\sin \theta_{K_1} + \sin \theta_{K_2}}{\cos \theta_{K_1} + \cos \theta_{K_2}}) \quad (13)$$

以及第二个弧的轨迹方程（当 $H$ 在弦 $FYOK_1$ - $FYOK_2$ 的下方时）为：

$$\rho_3 = (1 - \frac{\tan |\frac{\theta_{K_1} - \theta_{K_2}}{2}|}{\tan \alpha_3}) \cdot r \cdot \sqrt{2 + \cos(\theta_{K_1} - \theta_{K_2})} \cdot \cos(\theta - \arctan \frac{\sin \theta_{K_1} + \sin \theta_{K_2}}{\cos \theta_{K_1} + \cos \theta_{K_2}}) \quad (14)$$

#### Step4: 最终定位模型的构建

综合上述步骤，我们最终可以构建起**多点轨迹定位模型**实现多被动接收信号的无人机的定位：

$$\varphi = C_2^1 \cdot C_2^1 \cdot C_2^1 \quad (15)$$

其中， $\varphi$ 是一个集合，包含所有轨迹方程能够组合成的方程组；右边的三个 $C_2^1$ 分别表示从 $\alpha_1$ 、 $\alpha_2$ 、 $\alpha_3$ 所产生的两个轨迹方程中各挑选出的一个方程，三个 $C_2^1$ 相乘表示所挑选出的三个方程组合成的方程组（含三个方程）。最终集合 $\varphi$ 中共包含八个方程组，其中仅有唯一一组有解，即三个轨迹交于一点，该点即为对应无人机的定位。

最终的多点轨迹定位示模型意图如下所示：

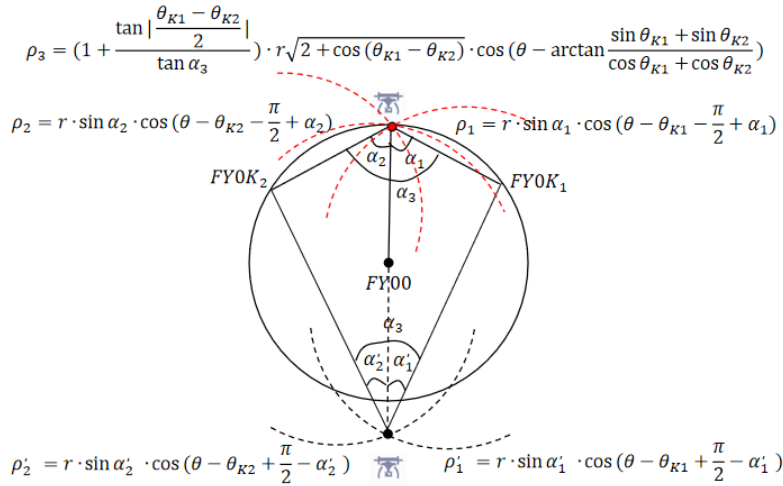


图 7. 无人机定位示意图

该图上方的红色虚线展示了其中一架无人机被定位的情况，三条红色虚线分别为其在夹角 $\alpha_1$ 、 $\alpha_2$ 、 $\alpha_3$ 下产生的三条弧轨迹，而最终三条轨迹的交点成功定位该无人机。图中被定位的无人机刚好位于圆上。在三架无人机发送信号的情况下，此定位模型可用于对任一无人机（无论圆上还是圆外圆内）的定位。



## 5.2 问题一第（2）问：基于改进信息源回溯与多点定位的无人机最优增量规划

第（2）问要我们求解能够实现有效定位的发送信号的无人机的增量。为了避免外界的干扰，应尽可能少向外发射电磁波信号。因此，增加的信息源无人机应该尽可能少。对于此，我们建立一个 0-1 规划模型，并设计出循环搜索算法求解。

在每一次的循环中，都涉及到求解轨迹方程并联立进行定位的过程，这既需要角度信息，又需要角度对应的两架信息源无人机编号（同第（1）问中的建模分析）。

但第（2）问的条件中，被动接收信号的无人机不知道发送信号的信息源无人机的编号，因此我们改进第（1）问中设计的信息源回溯算法，在匹配角度与编号信息之前先通过误差范围内的比较确定编号。并在此基础上探索增加多少架无人机能够实现有效定位。第（2）问流程如下图所示：

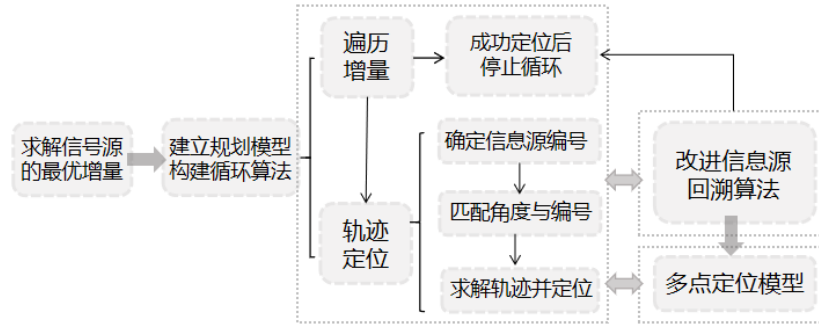


图 8. 问题一第（2）问流程图

### 5.2.1 模型的建立

基于对第（2）问的分析，我们建立基于 0-1 规划的无人机最优增量规划模型。决策变量：

$$\Delta x_i = \begin{cases} 0 & \text{第} i \text{次循环后不增加信号源} \\ 1 & \text{第} i \text{次循环后要增加信号源} \end{cases} \quad (16)$$

目标方程：

$$\min \sum_{i=1}^8 \Delta x_i \quad (17)$$

因为我们设计的循环搜索寻找其最小值，故该规划模型没有约束条件，直接进入循环搜索中。模型求解的循环流如下所示：

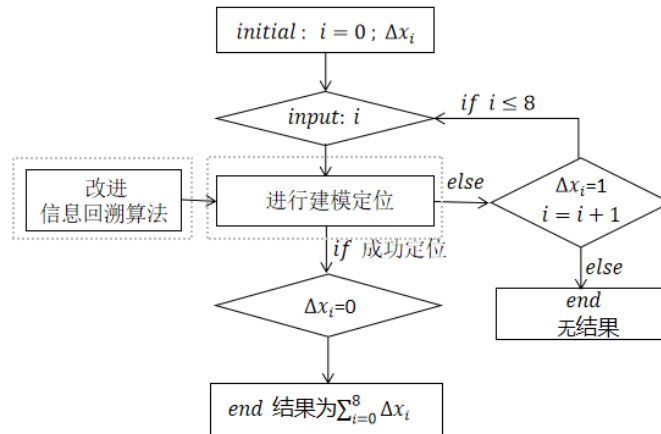


图 9. 模型求解流程图

### 5.2.2 改进信息源回溯算法

第（2）问与第（1）问最大的区别在于除了FY00与FY01之外，其他发送信号、位置无偏的无人机的编号未知，故我们不能直接利用第（1）问中编写的信息回溯算法对角度信息与编号进行匹配，而是需要先确定信息源无人机的编号，才能进行角度信息与对应编号的匹配。换言之，我们需要在进行匹配前加入确定无人机编号这一步骤。基于分析，我们在此改进我们的信息源回溯算法（Matlab 代码见附录3）：

**Step1:** 建立集合 $M$ ，储存十架无人机中任意三架组合所形成的所有夹角信息：

$$M = \{x | x \in \sum_{a,b,c \in F} \sum_{k=1}^3 T_{abc}^{(k)}\} \quad (18)$$

其中， $F = \{FY00, FY01, \dots, FY09\}$

其中， $T_{abc}^{(k)}$  表示 $a, b, c$ 三架无人机依次作顶点所形成的角度信息的集合。即： $k = 1$ 时，集合里的角度信息为 $\angle bac$ （以 $a$ 为顶点）； $k = 2$ 时，集合里的角度信息为 $\angle abc$ （以 $b$ 为顶点）； $k = 3$ 时，集合里的角度信息为 $\angle acb$ （以 $c$ 为顶点）。

**Step2:** 根据位置有偏、被动接收信号的无人机 $H$ 所获取的夹角信息 $\alpha$ ，在 $M$ 中找到近似的角度信息，并进行误差范围内的比对：

$$|x - \alpha| < \varepsilon \quad (19)$$

其中， $x$  为集合 $M$ 中的角度， $\varepsilon$  为集合 $M$ 中所有 $x$ 与 $\alpha$ 产生的差异中最小的差异（即最小误差），则满足该式的 $x$ 即为 $H$ 在其理想无偏位置的角度。

**Step3:** 上一步中找出的 $x$ 所对应的无人机的编号信息就对应为无人机 $H$ 所接收到的信号源的编号，即向 $H$ 发送信号的两架无人机的编号得以确定。

**Step4:** 确定发射信号的无人机编号之后，回到 5.1.2 中的算法步骤，将角度与无人机进行匹配。算法可视化如下图：

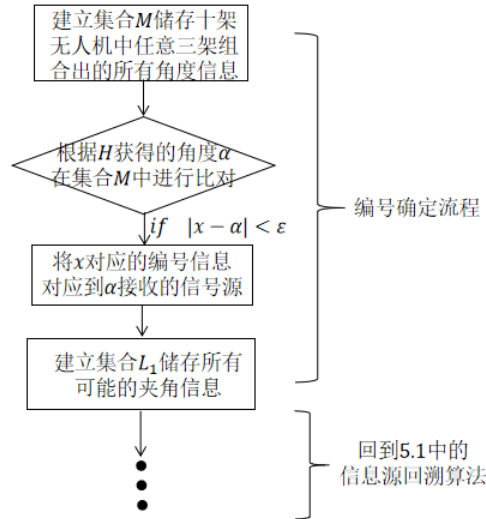


图 10. 改进的信息源回溯算法

注：以上改进的信息回溯算法有效的核心前提假设在于：

1. 位置有偏的无人机仅是略有偏差，这里我们根据测量学相关理论假设题目中的“略有偏差”的最大误差不超过 $5^\circ$ 。故我们可以通过接收到的角度信息与无偏差情况下的理想角度信息进行误差范围内的比对从而确定发射信号的无人机的编号。

2. 只有每次发射信号的与三架无人机均相邻近的无人机会进行信息回溯定位，非

邻近的情况如下图所示，若两架无人机的理想位置连线上存在其它无人飞机，则判定这两架无人机为非邻近。例如：FY05、FY04、FY01 作信号源发射信号时，FY03 不会进行定位；FY00、FY01、FY02 作信号源发射信号时，FY03 才会进行定位。

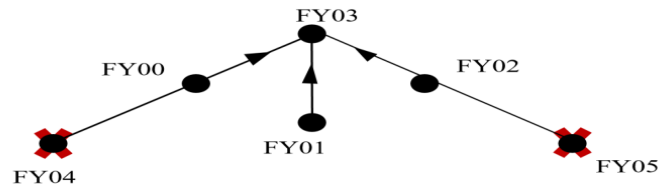


图 11. 改进的信息源回溯算法

3.若接受信号的无人机与发射信号的无人机位于同一条直线上，则会发生信号重叠，无法定位，此时无人机收到无效信号，如下右侧图中红色箭头所示，有效信号为左图所示。因此我们通过理想队形各架无人机的相对位置，选择可以与这发射信号的三架无人机构成四边形的无人机，进行位置调整。

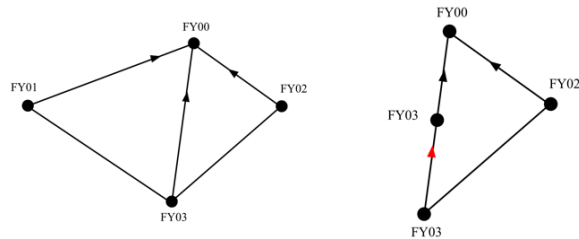


图 12. 改进的信息源回溯算法

值得注意的是，在比较过程中，会出现两个不同的信息源组合成相同角度的情况，这种情况下可能会对信息源的编号产生误判。但这种情况出现在信息源对称的时候，这时信息源编号的误判并不会对被动接收信号的无人机的定位产生影响。

以下我们举例分析：

我们不妨设FY03为被动接收信号、需要被定位的无人机。通过其他无人机分别与FY01、FY00向FY03发送信号产生的夹角来分析对称情况下夹角度数相同的情况。产生的所有角度信息如下表所示：

表 1. 有偏的无人机夹角信息汇总（以FY03接收信号为例）

信号源	与FY01产生的夹角	与FY00产生的夹角
FY09	$\angle 931 = 20^\circ$	$\angle 930 = 30^\circ$
FY08	$\angle 831 = 40^\circ$	$\angle 830 = 10^\circ$
FY07	$\angle 731 = 60^\circ$	$\angle 730 = 10^\circ$
FY06	$\angle 631 = 80^\circ$	$\angle 630 = 30^\circ$
FY05	$\angle 531 = 100^\circ$	$\angle 530 = 50^\circ$
FY04	$\angle 431 = 120^\circ$	$\angle 430 = 70^\circ$
FY02	$\angle 231 = 20^\circ$	$\angle 230 = 20^\circ$
FY01	不成角	$\angle 130 = 50^\circ$

注：表中用数字简化角的表示。例如， $\angle 931$ 为信号源FY09与FY01向FY03发送信号形成的夹角。由表可见，不同信号源下产生了相同度数的夹角，比如， $\angle 930$ 与 $\angle 630$ 角度相同； $\angle 130$ 与 $\angle 530$ 角度相同等。情况如下图所示：

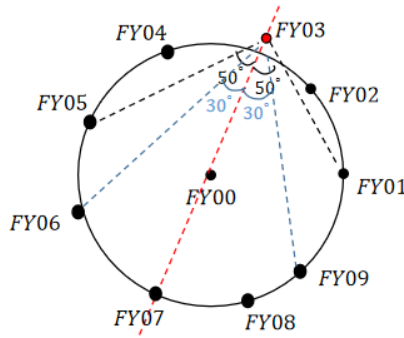


图 13. 对称情况示意图（以FY03接收信号为例）

由图可见，这些相等角所对应的信息源无人机均与过FY00与FY03的直径所在的直线（即图中红色虚线）相对称。此时，若发生误判，如将FY01误判为FY05，这并不会影响对于FY03的定位，因为最终求得的该角对应的轨迹方程也会与红色虚线对称，在这种情况下，轨迹方程最终同样交于同一点，即仍能够准确定位FY03。

因此，由于对称角相同问题引起的编号误判并不会影响最终定位，故我们不用考虑该误判是否发生。

### 5.2.3 模型的求解与结果分析

#### （1）求解结果：

在我们循环遍历（图 10）之后，得到最优解  $\min \sum_{i=1}^8 \Delta x_i = 1$ ，即只用增加 1 量无人机发射信号就可以实现对被动接收信号无人机H的有效定位。

#### （2）结果分析：

从求解的过程来看，当  $\sum_{i=1}^8 \Delta x_i = 1$  时，此时共有三架无人机作为信息源向被动接收信号的无人机发送信号，只是其中一架编号未知（另外两架为题目给定的FY00与FY01）。而在我们改进信息源回溯算法的帮助下，该架新增发送信息的无人机编号可以确定下来，从而问题（2）变成和问题（1）相同的情景，则被动接收信号的无人机一定可以被有效定位，我们的循环算法也就此停止。

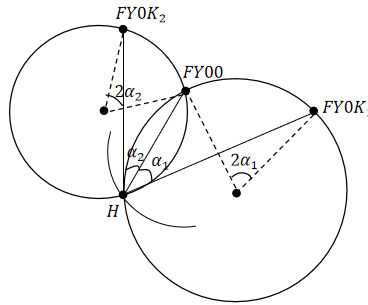


图 14. 增加一架信号源无人机后有效定位示意图

### 5.3 问题一第（3）问：基于贪婪思想的定位寻优模型

分析题目所给的无人机初始位置表可知，除了FY00与FY01位置正确以外，其他八架无人机位置均有偏。针对于此我们建立定位寻优模型。

我们先选取原始位置与理想无偏位置偏差最小的无人机（即FY02）作为除FY00与FY01外的第三个信息源，与FY00、FY01一起向其他无人机发送信号确定其位置，并进行第一次迭代，迭代完成后选取误差最小的点作为新的信息源进行下一次迭代。如此往复，在多次迭代后，最终各无人机的调整位置无限收敛于理想无偏位置，即



**情况二：** FY06被动接受信号时，如图 17 所示：

在这种情况下， $x_1 + x_2$ 为定值： $x_1 + x_2 = \frac{16\pi}{9}$ ；同样由正弦定理，得到方程组：

$$\begin{cases} \frac{100}{\sin \alpha_1} = \frac{l}{\sin(\pi - x_1 - \alpha_1)} \\ \frac{100}{\sin \alpha_2} = \frac{l}{\sin(\pi - x_2 - \alpha_2)} \\ x_1 + x_2 = \frac{16\pi}{9} \end{cases} \quad (22)$$

由牛顿迭代法求出 $x_1$ 、 $x_2$ ，接着可以结合图形并利用正弦定理求解 $H(\rho, \theta)$ ：

$$\rho = \frac{100 \cdot \sin(\pi - x_1 - \alpha_1)}{\sin \alpha_1}; \quad \theta = -x_2 \quad (23)$$

以上两种情况为以FY02作为第三个信号源发射信号时出现的情况，在 Step2（下述）的无人机位置迭代调整以后，我们会选取新的信号源，此时会产生新的讨论情况，但受论文篇幅限制，此处不再详细讨论，讨论思路与上述两种情况一致。

### Step2：对近似定位的无人机位置进行迭代调整

为了阐述迭代的思路，此处以 Step1 分析中的情况一为例， $H$ 的坐标为：

$$\rho = \frac{100 \cdot \sin x_1}{\sin \alpha_1}; \quad \theta = -(\pi - \alpha_2 - x_2 - \frac{7\pi}{18})$$

因为FY02实际的位置是有偏的，但在计算中，我们假设其处于无偏位置，并利用无偏位置下得到的数据计算出被动接收信号的无人机 $H$ 的坐标。因此我们计算得到的 $H$ 的坐标也是与其实有偏的，所述关系可以如下图所示：

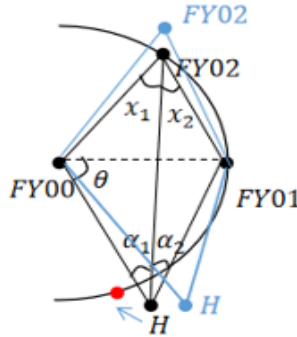


图 18. 偏离情况与迭代过程演示

图中蓝色点 $H$ 与蓝色点FY02为其实际的位置，但无人机 $H$ 会将FY02视作位置无偏的情况（即图中的黑色点FY02），在黑色点的情况下，与FY00、FY01向 $H$ 发送信号定位得到的是与实际位置有偏的 $H$ ，即黑色点 $H$ 。在该定位情况下，让定位有偏的黑色点 $H$ 向其理想位置点（红色点）移动（蓝色箭头表示黑色点 $H$ 向理想位置移动的过程），但实际位置中的蓝色点 $H$ 随着这个移动过程（蓝色箭头）移动，最终并不能抵达其理想位置（红色点），但是其真实位置与其理想状态的差距会减小。

这样通过近似定位与朝理想位置移动的过程就为一次迭代过程。每一次迭代都会进一步缩小实际位置与理想位置的差距，经过多次迭代后，差距不断缩小，真实位置不断向理想位置收敛，当收敛到一定程度，即误差足够小时，我们近似认为其抵达了理想的位置（即该点抵达了其圆形编队中的位置）。

由于该迭代过程中每一次我们都选取与实际距离误差最小的无人机作为下一次迭



代过程的信号源，这一思想其实是贪婪思想，即只注重当前最优的情况，故上述建立的模型为**基于贪婪思想的定位寻优模型**。

### 5.3.2 模型的求解

模型的求解过程主要是对无人机定位后进行迭代调整的过程，我们设计以下的迭代算法进行求解（Matlab 代码见附录 5、6、7）：

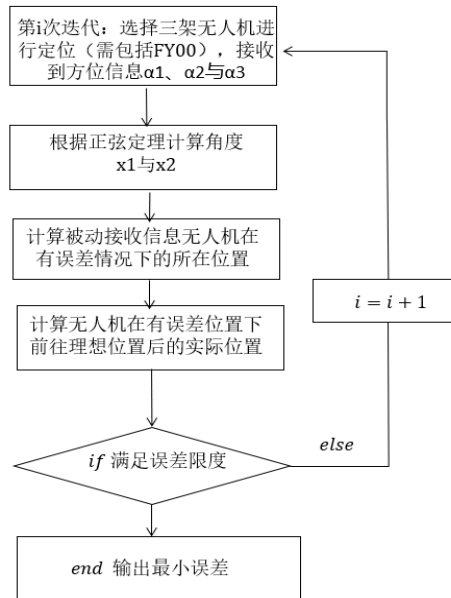


图 19. 迭代流程图

首先，基于题目给出的十架无人机的初始位置信息，我们通过 Matlab（代码见附录 4）绘制出所有无人机的初始位置与其理想位置的示意图：

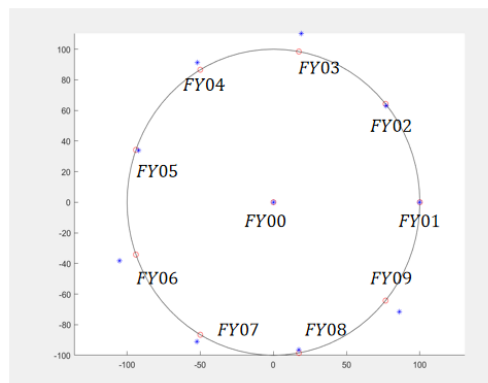


图 20. 各无人机的初始位置

其中，圆上的红色空心圆表示各无人机的理想位置；蓝色点表示各无人机的实际位置。可以看出，初始位置中除了FY00和FY01没有偏差以外，其余八架无人机均有较大偏差。其中偏差最大的几架无人机为FY03、FY06、FY09。

接下来我们先以FY02作为第三个信息源无人机发送信号，并开始迭代：

表 2. 第一次迭代结果

	FY02	FY03	FY04	FY05	FY06	FY07	FY08	FY09
$x$	76.9623	13.2557	-50.126	-92.8944	-93.4266	-51.3673	20.8285	80.9321
$y$	63.1241	96.5947	83.2211	33.535	-33.9651	-88.7565	-101.0852	-61.3664
$d$	1.20889	4.52129	3.38375	1.26503	0.59215	2.55132	4.33360	5.21642

其中， $d$ 表示迭代之后，每架无人机的实际位置距其理想位置的差距（差距计算的 Matlab 代码见附录 8）。

第一次迭代之后，我们发现FY06的实际位置距其理想位置差距大大缩小（表中标红的数据），因此我们将FY06替换FY02作为第三个信息源发送信号，并进行各无人机调整位置的第二次迭代。

表 3. 第二次迭代结果

	FY02	FY03	FY04	FY05	FY06	FY07	FY08	FY09
$x$	76.5466	17.1455	-49.986	-93.4933	-93.4266	-50.0054	15.799	76.3405
$y$	64.313	98.3329	86.1775	33.9298	-33.9651	-85.9862	-92.9819	-64.4687
$d$	0.06716	0.26451	0.42524	0.54833	0.59215	0.61632	5.71748	0.32512

第二次迭代后，可见各架无人机实际位置逐渐向其理想位置靠近。此时，FY03距离其理想位置最近，因此我们将FY03作为新的信息源对各无人机的位置调整进行第三次迭代。

表 4. 第三次迭代结果

	FY02	FY03	FY04	FY05	FY06	FY07	FY08	FY09
$x$	76.8107	17.1455	-49.9995	-93.9638	-94.066	-50.0019	17.6417	76.9563
$y$	64.1606	98.3329	86.44500	34.1988	-34.2576	-86.853	-98.6529	-64.0831
$d$	0.23776	0.26451	0.157500	0.00636	0.11154	0.25051	0.3260	0.40266

三次迭代完成以后，所有无人机实际位置不断向理想位置收敛，其差距已经全都缩小到 1m 以内，此时我们认为其，迭代停止。每次迭代的效果图如下：

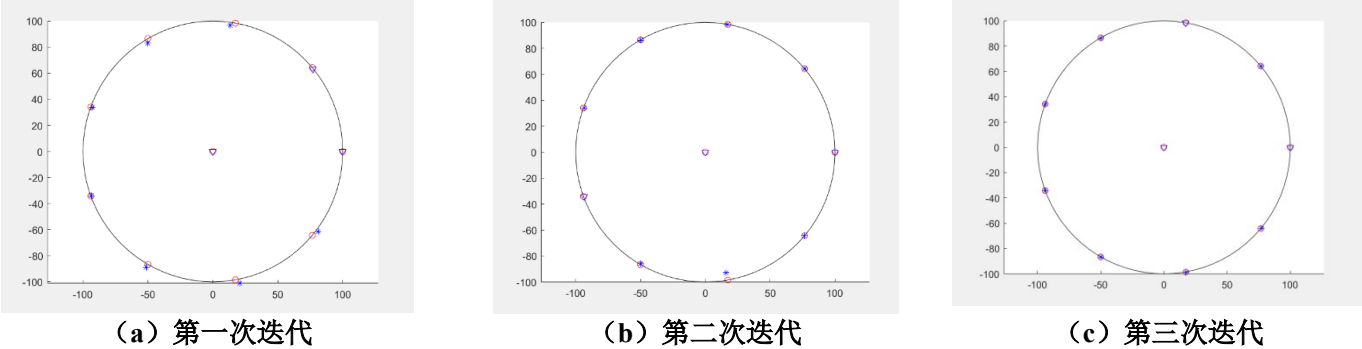


图 21. 每次迭代的效果图

从三次迭代的效果图中，我们可以看见无人机的实际位置向理想位置收敛的过程，直至第三次迭代，无人机实际位置几乎与理想位置重合，说明通过我们的定位寻优模型给出的无人机调整方案效果良好。

为了进一步检验模型的效果，我们计算每次迭代完成后十架无人机实际位置与理想位置的平均偏差距离如下表：

表 5. 迭代次数与平均偏差距离

	迭代前	第一次迭代	第二次迭代	第三次迭代
平均偏差距离	5.128424326	2.307245791	0.855633382	0.175687179

并绘制平均偏差距离与迭代次数的图像如下：



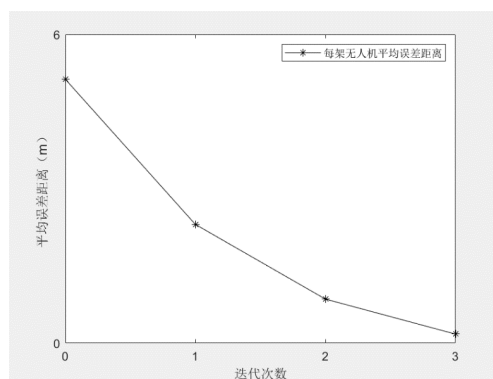


图 22. 平均偏差距离随迭代次数增加的变化

由该图可见，随着迭代次数的增加，十架无人机实际位置与理想位置的偏差的平均值不断减小，其中第一次迭代所产生的收敛作用最为明显。随着继续迭代的进行，其收敛速度逐渐减慢，直至第三次迭代结束，十架无人机的平均误差已经急剧逼近于零，继续迭代的效果不再明显，故最终我们总共进行三次迭代。

上述三次迭代的过程（即十架无人机不断向理想位置靠近的过程）即为问题一第（3）问的队形调整方案。

#### 5.4 问题二

分析题目所给的锥形无人机编队可知，我们仍然可以采用问题一第（1）问所建立的定位模型与第（3）问的迭代算法进行无人机位置调整，从内往外逐层调整，而后可以总结到最基础的四边形无人机编队，发散到更多其他能够应用于实际场景的无人机编队，总结调整方案。问题二流程图如下：

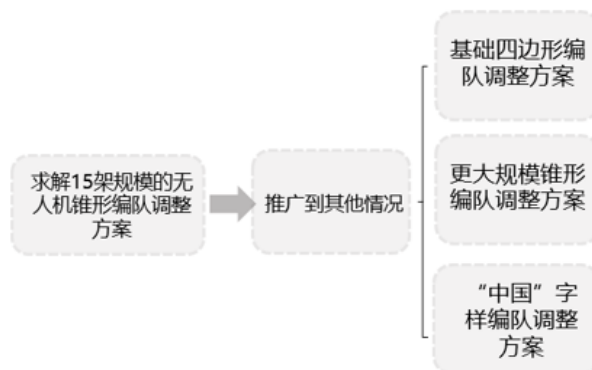


图 23. 问题二流程图

##### 5.4.1 锥形编队调整方案的求解

由于锥形队形实则为等边三角形组合成的编队队形。根据 5.3 节中方案的思想，我们可以对锥形编队使用多动点轨迹定位模型与信息源回溯算法来辅助无人机进行定位调整。我们将在锥形队形中的 15 架的无人机重新编号，例如：左下角的无人机为 1 号，坐标默认为(0,0)，如图所 23 所示。

在锥形编队中，第一次使用中心的三架无人机发送信号，即如上图所示的编号为 6、7、10 号的无人机。可以发现这三架无人机与周围无人机连接形成一个圆内切正六边形，所以我们同理利用 5.3 节的方法对同一圆上的点进行迭代位置调整。如图 24 所示。

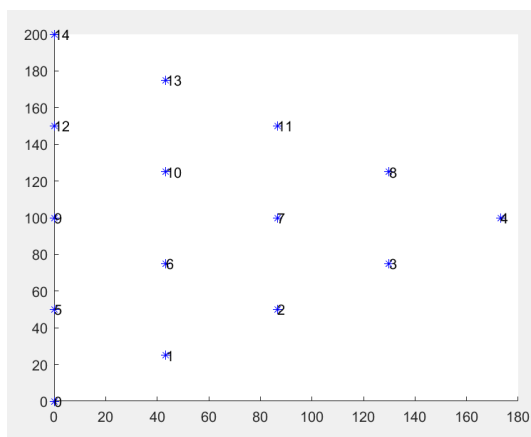


图 24. 锥形编队编号图

同理 5.3 节方案中的步骤一，首先选取 6、7、10 号无人机发射信号，使用多动点定位模型，如图 24 三个圆周上的其余无人机进行定位调整。然后同理 5.3 第二三步骤进行多次迭代，每次除作为基准的无人机外，更换其余两架发射信号的无人机，直到达到误差足够小时停止。此时完成了 1、2、3、5、8、9、11、12、13 号无人机的定位调整。

接下来，轮流使与锥形三个顶点相邻的两架无人机与另一架未被这两架无人机遮挡信号的无人机发射信号，上述四架无人机形成了圆的内切等腰梯形，如图 25 所示。继续进行迭代调整，最终使顶点上的无人机也调整到预期位置。

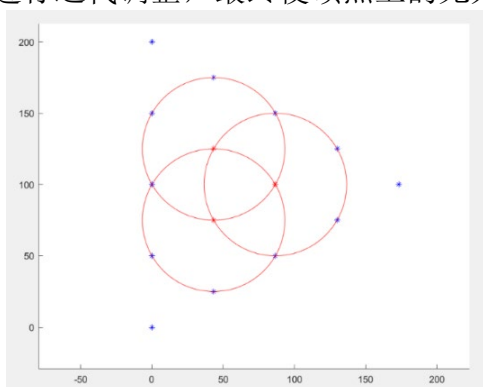


图 25

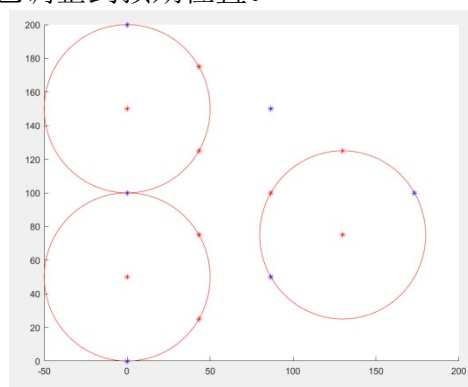


图 26

对于无人机数量超过 15 的锥形编队，我们上述的思路进行定位调整，即若锥形编队最宽处无人机数为奇数，则从最中心等边三角形顶角的三架无人机开始发射信号；若锥形编队最宽处无人机数为偶数，则次中心等边三角形顶角无人机首先发射信号，如图例所示：

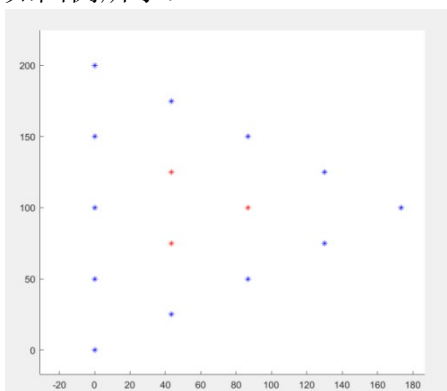


图 27. 奇数情况发射信号无人机示意

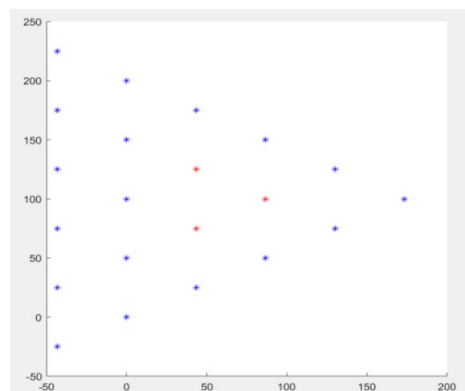


图 28. 偶数情况发射信号无人机示意

发射信号后进行多次迭代，调整外层无人机之后，再按照相同方法每次指定三架无人机发射信号，再向外一层调整该层无人机位置。直到达到最外层，调整结束。

#### 5.4.2 其他编队的模型建立与求解

我们将问题一中已经建立的模型归纳，推广到任意四架及四架以上，且队形中存在由四架无人机构成的四边形的无人机队形上。

考虑到信息源回溯算法中信号来源判断的准确性，我们每次仅允许三架无人机发射信号，与这三架无人机相邻的无人机进行位置的调整。

**Step1:** 对于任意超过四架无人机的队形，在所有无人机位置离理想位置都是略有偏差的假设下，我们首先规定一架无人机作为基准点，再从系统中选择与之相邻的两点作为信号发射源。

考虑到初始点都可能有偏差，利用问题二第三小问中的迭代算法，建立数量大于四架的无人机群（其中任意三架无人机不会连成一条线）的位置迭代调整算法。接下来，我们考虑最简单的情况（即初始进行位置迭代调整的无人机仅有四架）利用此算法进行这四架无人机的位置调整,示意图如下：

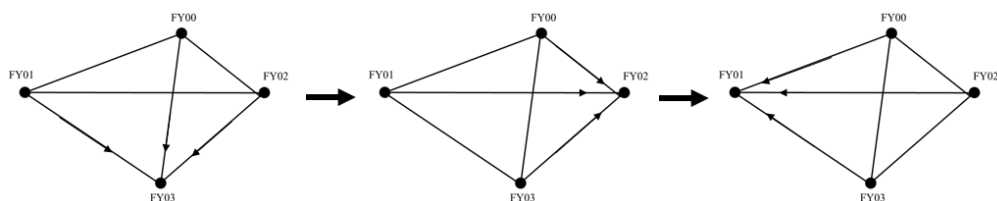


图 29. 四边形编队位置调整示意图

在这步中，选取 FY00 为基准点，其余三架无人机以 FY00 为中心，循环选取两架无人飞机为信号发射源，剩余一架飞机接收信号，实现自身定位后进行方位调整。

**Step2:** 在初始的四个点调整到误差大小可以接受的范围之内之后，可以近似认为这四个点这四架无人机的位置已经无偏差。然后，从四架无人机中选择三架无人机作为信号发射源，对其余满足信息回溯算法假设的无人机实现有效精准定位，一步到位进行方位调整。

**Step3:** 每次从已调整好的所有无人机中，任意选择可还未调整好位置的无人机提供有效信号的三架无人机发射信号，相邻且收到有效信号的无人机，可以进行精确的位置调整。重复上述步骤，最终可以实现所有点的有效调整。

综上所述，上述方法适用于任意队形无人机半自动化的位置调整。然而上述方法可以通过人为的干预设计更加简洁的调整方案。对于对称图形，例如 5.4.1 节中锥形模型的调整方法，我们可以利用同一圆上的无人机可以同时通过迭代算法进行误差调整，从而减少队形调整所使用的时间。

下面，我们将继续通过一个例子来说明如何有效设计方案，减少无人机调整的步骤，实现更优的调整方法。

#### 5.4.2 其他编队的模型建立与求解

题目所提供的两个队形，均为特殊的对称图形（圆形编队与锥形编队）。但如 5.4.2 中所示的非对称图形，我们可以通过有效的方案设计，实现较短步骤内的有效调整。考虑到更普适的队形调整方案，我们在此设计了“中国”字样的队形，并利

用 5.4.2 中设计的方案对该队形进行调整方案的设计。

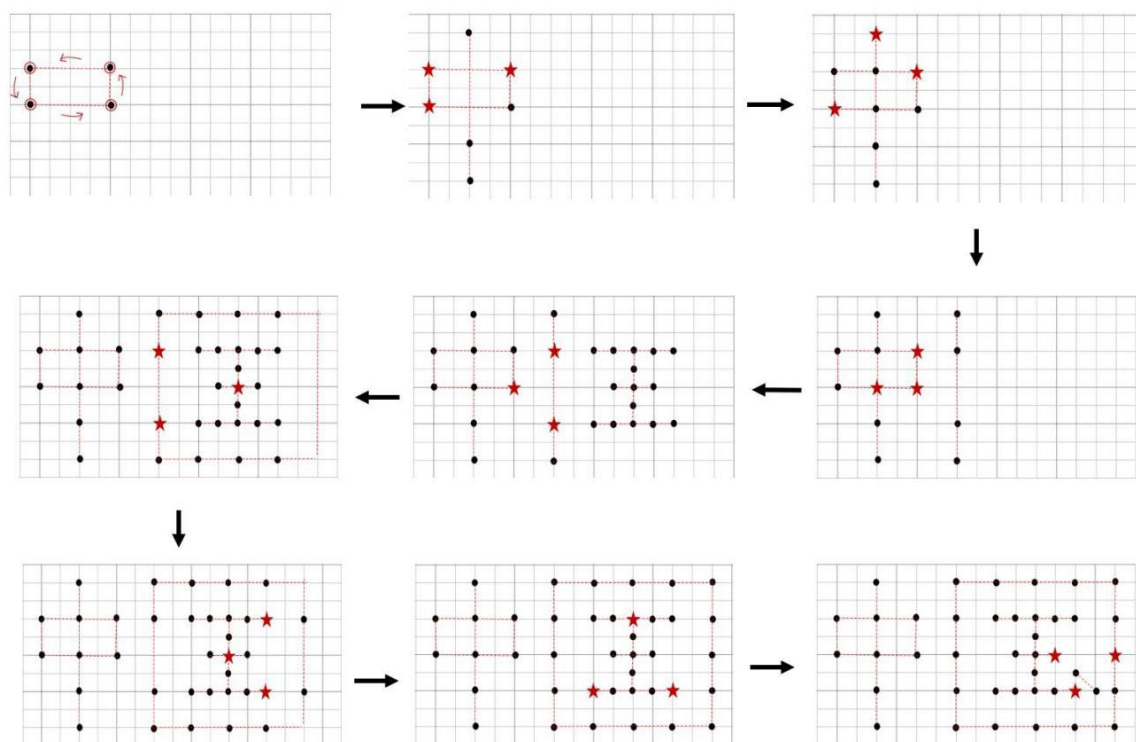


图 30. “中国”队形调整方案

注：上图中每幅小图中的三个星形图标为此步骤中发射信号的三架无人机，小图中出现的新的点出现，表示经过此步骤后该点所代表的无人机可以精确调整自身的位置。

上图展示了“中国”队形的调整方案，该方案通过九步实现“中国”队形的有效调整。第一步，选取四架无人机实现循环迭代定位。接下来，以这四架无人机为基础，对其余无人机精准定位后，进行精确的方位调整。

## 六、模型的分析与检验

本文主要针对问题一第（2）问的模型进行模型的稳健性分析（**Matlab 代码见附录 10**）。由于本题设置条件略有偏差，偏离角度非常小，因此讨论偏差角度不同的情况下，信息源回溯算法表现的稳定性与敏感性。

因此我们随机挑选几架被动接受信号进行调整定位的无人机。随机选则 FY03、FY05、FY06 进行讨论。由题意已知发射信号的无人机有 FY01 与 FY00,现在通过改进的信号源回溯算法进行第三架未知信号的无人机查询，在稳健性分析中我们预先假设该机编号为 FY02。

在稳健性分析中，我们使用的方向角误差范围为 $0.01rand \sim 1rand$ （换算成角度制为 $0.573^\circ \sim 5.73^\circ$ ）。来对 FY03 寻找未知发射信号源无人机编号结果准确性测试。

通过对不同方向角误差情况下进行信息源回溯算法的应用，可以得到单次进行信息源回溯算法能得到结果所对应的误差角度与其他进行信息源回溯算法不能得到结果的误差角度，如下图所示：

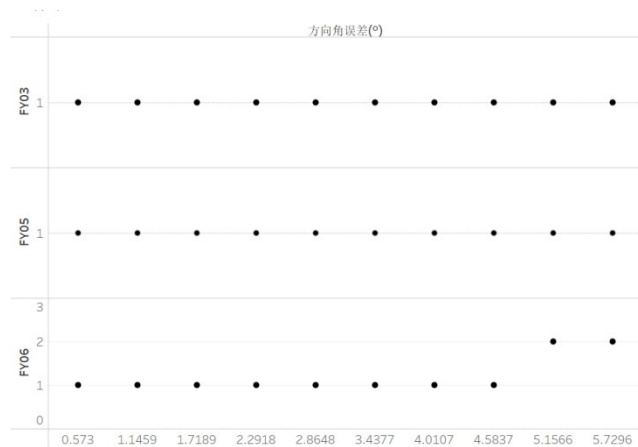


图 31. 误差角度图

通过该图可知对于 FY03、FY05 来说可以完全适配应用场景，而对于 FY06 来说则只有最后两个不符合情况，经计算  $5^\circ$  之内符合要求，因此证明了我们对于“略有偏差”的角度定义是正确的，同时也得出了我们的改进信息源回溯算法在本文场景中具有较高的准确率、稳健性与较好的实用性。

## 七、模型的评价、改进与推广

### 7.1 模型的优点与缺点

#### 优点：

(1) 本文的第一问信息源回溯算法具有准确性的保证，可以保证进行多次筛选循环得到确定发射信号无人机组合的情况下，各方位角形成的无人机组合。而对于改进信息源回溯模型，在略有偏差的情况下也能保证良好的准确率；

(2) 本文在问题一中研究的多点轨迹定位模型适用于其他只能接受方位信息无人机且编队拥有四边形的使用场景，因此可以应用于大量其他有电磁静默需求的无人机队形编队。

#### 缺点：

本文无源定位模型有一定的局限性，其中之一体现在模型对于编队中任意无人机连线不存在四边形的队形无法进行精确定位与调整，需要引入距离测量或者四边形的编队结构。此外，本文无源定位后队形调整迭代方案仍存在优化的空间，可以进行更多情况的讨论。

### 7.3 模型的改进与推广

在模型进一步的改进中，我们应该考虑到模型表达的完整性与求解的可操作性，在合适的地方使用直角坐标或者极坐标等数学表示。同时模型改进也可以侧重于无人机初始误差的角度不仅限于“略有误差”，可以朝误差较大的情况进行建模，或者在三维空间的情况下仅靠方向信息进行队形调整。

## 八、参考文献

- [1] 吕林根,许子道. 解析几何（第五版）[M]. 北京：高等教育出版社，2019：第二章第1节.
- [2] 邱华鑫,段海滨,范彦铭.基于鸽群行为机制的多无人机自主编队[J].控制理论与应用,2015,32(10):1298-1304.

## 附录

### 附录 1

介绍： 支撑材料的文件列表

1. Q1A1.m, 第一题第 (1) 小问源程序
2. Q1A2.m, 第一题第 (2) 小问源程序
3. Q1A31.m, 第一题第 (3) 小问第一次迭代源程序
4. Q1A32.m, 第一题第 (3) 小问第二次迭代源程序
5. Q1A33.m, 第一题第 (3) 小问第三次迭代源程序
6. Q1A3\_calerror.m, 第一题第 (3) 小问计算误差源程序
7. plot\_1.m, 第一题第 (3) 小问作图代码
8. plot2.m, 第二题与稳健性分析作图代码
9. 第三问迭代过程与结果.xlsx
10. iter.m, 第一题第 (3) 小问有偏差位置移向理想位置计算函数

### 附录 2

介绍： 第一题第 (1) 小问信息源回溯算法

```
t=0:2*pi/9:2*pi;
x=[0 100*cos(t)];y=[0 100*sin(t)];
k=5;%有偏差无人机的编号 (FY00、FY03、FY01除外),假设为5,已知编号为FY00、FY03、FY01
B=[x(k+1),y(k+1)];
l2=[];
l1=[[0,1];[0,2];[0,3];[0,4];[0,6];[0,7];[0,8];[0,9];[1,2];[1,3];[1,4];[1,6];[1,7];[1,8];[1,9];[2,3];[2,4];[2,6];[2,7];[2,8];[2,9];[3,4];[3,6];[3,7];[3,8];[3,9];[4,6];[4,7];[4,8];[4,9];[6,7];[6,8];[6,9];[7,8];[7,9];[8,9]];
for i=1:36
    M=l1(i,:);
    m=M(1)+1;n=M(2)+1;
    C=[x(m),y(m)];
    A=[x(n),y(n)];
    a=sqrt((C(1)-B(1))^2+(C(2)-B(2))^2);
    b=sqrt((A(1)-C(1))^2+(A(2)-C(2))^2);
    c=sqrt((A(1)-B(1))^2+(A(2)-B(2))^2);
    cosB=(a^2+c^2-b^2)/(2*a*c);
    l2=[l2 cosB];
end
alpha1=pi/18+0.1;%输入有偏差无人机接收到的FY00与未知编号无人机的夹角信息
alpha2=2*pi/9+0.001;%输入有偏差无人机接收到的FY00与未知编号无人机的夹角信息
alpha3=5*pi/18+0.07;%输入两架有偏差无人机接收到的FY00与未知编号无人机的夹角信息
C1=abs(l2-cos(alpha1))
C1_min=min(C1)
[row, col1]=find(C1==C1_min);
m1=l1(col1,:)%输出alpha1信号源两架无人机的编号 (无顺序)
```

```

C2=abs(l2-cos(alpha2));
C2_min=min(C2);
C2_min=6.431705247449404e-04;%由min(C2)得出结果，忽略MATLAB的数字显示问题
[row, col2]=find(C2==C2_min);
m2=l1(col2,:)%输出alpha2信号源两架无人机的编号（无顺序）
C3=abs(l2-cos(alpha3));
C3_min=min(C3);
[row, col3]=find(C3==C3_min);
m3=l1(col3,:)%输出alpha3信号源两架无人机的编号（无顺序）

```

### 附录 3

介绍： 第一题第（2）小问改进信息源回溯算法

```

t=0:2*pi/9:2*pi;
x=[0 100*cos(t)];y=[0 100*sin(t)];
k=3;%有偏差无人机的编号（FY00、FY01、FY0k除外）,在这里假设情况为3,已知编号FY00、FY01
B=[x(k+1),y(k+1)];
l3=[];l4=[];
l1=[[0,2];[0,4];[0,5];[0,6];[0,7];[0,8];[0,9]];
l2=[[1,2];[1,4];[1,5];[1,6];[1,7];[1,8];[1,9]]; %每架无人机有自己独一无二的l1与l2
for i=1:7
    M=l1(i,:);
    m=M(1)+1;n=M(2)+1;
    C=[x(m),y(m)];
    A=[x(n),y(n)];
    a=sqrt((C(1)-B(1))^2+(C(2)-B(2))^2);
    b=sqrt((A(1)-C(1))^2+(A(2)-C(2))^2);
    c=sqrt((A(1)-B(1))^2+(A(2)-B(2))^2);
    cosB=(a^2+c^2-b^2)/(2*a*c);
    angle=acos(cosB);
    reangle=round(angle*180/pi);
    l3=[l3 reangle];
end
for i=1:7
    M=l2(i,:);
    m=M(1)+1;n=M(2)+1;
    C=[x(m),y(m)];
    A=[x(n),y(n)];
    a=sqrt((C(1)-B(1))^2+(C(2)-B(2))^2);
    b=sqrt((A(1)-C(1))^2+(A(2)-C(2))^2);
    c=sqrt((A(1)-B(1))^2+(A(2)-B(2))^2);
    cosB=(a^2+c^2-b^2)/(2*a*c);
    angle=acos(cosB);
    reangle=round(angle*180/pi);
    l4=[l4 reangle];
end
alphax=7*pi/18+0.07;%输入有偏差无人机接收到的FY00与未知编号无人机的夹角

```



信息

alphy=pi/9+0.005;%输入有偏差无人机接收到的FY01与未知编号无人机的夹角信息

alphax=round(alphax\*180/pi/10)\*10;alphy=round(alphy\*180/pi/10)\*10;

[row, X1]=find(l3==alphax);

[row, X2]=find(l3==alphy);

if isempty(X1)==0

L4=l4(:,X1);

if isempty(find(L4==alphy))==0

c1=X1(find(L4==alphy));

m1=l1(c1,:)%输出信号源FY00与未知编号无人机的编号（无顺序）

m2=l2(c1,:)%输出信号源FY01与未知编号无人机的编号（无顺序）

end

else

L4=l4(:,X2);

if isempty(find(L4==alphax))==0

c2=X2(find(L4==alphax));

m1=l1(c2,:)%输出信号源FY00与未知编号无人机的编号（无顺序）

m2=l2(c2,:)%输出信号源FY01与未知编号无人机的编号（无顺序）

end

end

#### 附录 4

介绍： 第一题第（3）小问作图代码

%理想位置

b=[0 0

100 0

76.6044 64.2788

17.3648 98.4808

-50 86.6025

-93.9693 34.202

-93.9693 -34.202

-50 -86.6025

17.3648 -98.4808

76.6044 -64.2788];

scatter(b(:,1)',b(:,2)','ro')

hold on

alpha=0:pi/40:2\*pi;

r=100;

x=r\*cos(alpha);

y=r\*sin(alpha);

plot(x,y,'k-');

axis equal

hold on

%初始位置

```

% c=[0 0
% 100 0
% 76.9623 63.1241
% 19.0442 110.369
% -52.1027 91.1609
% -92.0077 33.7429
% -105.2723 -38.2328
% -52.3889 -90.9967
% 17.3038 -96.4602
% 86.1478 -71.5721];
% scatter(c(:,1),c(:,2),'b*')
% hold on

%第一次迭代后的不发信号无人机的位置
% c=[13.2557 96.5947
% -50.126 83.2211
% -92.8944 33.535
% -93.4266 -33.9651
% -51.3673 -88.7565
% 20.8285 -101.0852
% 80.9321 -61.3664];
% scatter(c(:,1),c(:,2),'b*')
% hold on

%第一次迭代后的发信号无人机的位置
% c=[0 0
% 100 0
% 76.9623 63.1241];
% scatter(c(:,1),c(:,2),'bv')
% hold on

%第二次迭代后的不发信号无人机的位置
% d1=[76.5466 64.313
% 17.1455 98.3329
% -49.9856 86.1775
% -93.4933 33.9298
% -50.0054 -85.9862
% 15.799 -92.9819
% 76.3405 -64.4687];
% scatter(d1(:,1),d1(:,2),'b*')
% hold on
%第二次迭代后的发信号无人机的位置
% d2=[0 0
% 100 0
% -93.426 -33.9651];
% scatter(d2(:,1),d2(:,2),'bv')
% hold on

%第三次迭代后的发信号无人机的位置

```

```

d2=[0 0
    100 0
    17.1455 98.3329];
scatter(d2(:,1),d2(:,2),'bv')
hold on

%第三次迭代后的不发信号无人机的位置
d1=[76.8107 64.1606
    -49.9995    86.445
    -93.9638    34.1988
    -94.066 -34.2576
    -50.0019    -86.853
    17.6417 -98.6529
    76.9563 -64.0831];
scatter(d1(:,1),d1(:,2),'b*')
hold on

%无人机平均误差距离折线图
x=0:1:3;%x轴上的数据，第一个值代表数据开始，第二个值代表间隔，第三个值代表终止
a=[5.128424326,2.307245791,0.855633382,0.175687179]; %a数据y值
plot(x,a,'-k'); %线性，颜色，标记
axis([0,3,0,6]) %确定x轴与y轴框图大小
set(gca,'XTick',[0:1:3])
set(gca,'YTick',[0:6:6])
legend('每架无人机平均误差距离');
xlabel('迭代次数') %x轴坐标描述
ylabel('平均误差距离（m）') %y轴坐标描述

```

## 附录 5

### 介绍： 第一题第（3）小问第一次迭代代码

```

clear;clc
t=0:2*pi/9:2*pi;
x=[0 100*cos(t)];y=[0 100*sin(t)];
C=[112*cos(80.21*pi/180),112*sin(80.21*pi/180)];
C=[105*cos(119.75*pi/180),105*sin(119.75*pi/180)];
C=[98*cos(159.86*pi/180),98*sin(159.86*pi/180)];
%C=[112*cos(199.96*pi/180),112*sin(199.96*pi/180)];
%C=[105*cos(240.7*pi/180),105*sin(240.7*pi/180)];
C=[98*cos(280.17*pi/180),98*sin(280.17*pi/180)];
%C=[112*cos(320.28*pi/180),112*sin(320.28*pi/180)];

A=[x(1),y(1)];B=[x(2),y(2)];D=[98*cos(40.1*pi/180),98*sin(40.1*pi/180)];
a=sqrt((D(1)-B(1))^2+(D(2)-B(2))^2);
b=sqrt((A(1)-D(1))^2+(A(2)-D(2))^2);
c=sqrt((A(1)-B(1))^2+(A(2)-B(2))^2);
cosC=(a^2+b^2-c^2)/(2*a*b);
d=70*pi/180

```

```

a=sqrt((C(1)-B(1))^2+(C(2)-B(2))^2);
b=sqrt((A(1)-C(1))^2+(A(2)-C(2))^2);
c=sqrt((A(1)-B(1))^2+(A(2)-B(2))^2);
c1=c;
cosC=(a^2+b^2-c^2)/(2*a*b);
aa1=acos(cosC)
a=sqrt((C(1)-D(1))^2+(C(2)-D(2))^2);
b=sqrt((A(1)-C(1))^2+(A(2)-C(2))^2);
c=sqrt((A(1)-D(1))^2+(A(2)-D(2))^2);
c2=c;
cosC=(a^2+b^2-c^2)/(2*a*b);
aa2=acos(cosC)
a=sqrt((C(1)-D(1))^2+(C(2)-D(2))^2);
b=sqrt((B(1)-C(1))^2+(B(2)-C(2))^2);
c=sqrt((B(1)-D(1))^2+(B(2)-D(2))^2)
c3=c;
cosC=(a^2+b^2-c^2)/(2*a*b);
aa3=acos(cosC);
m=[aa1,aa2,aa3];
cc=[c1,c2,c3];
a3=max(m);
c3=cc(find(m==a3));
xxx=find(m==a3);
[ad i]=setdiff(m,a3);
cc(xxx)=[];
m=m(sort(i))
a1=m(1);a2=m(2);
c1=100;c2=68;c3=100;
syms x;
f=c1*sin(a1+x)/sin(a1)-c2*sin(a2+d-x)/sin(a2);
p0=0;
tol=0.0000001;
maxK=1000;
[p,k,Y]=NTM(f,p0,tol,maxK);
fprintf("迭代值如下: ");
disp(Y);
x=Y(end)
y=d-x

%点6特殊情况
t=0:2*pi/9:2*pi;
x=[0 100*cos(t)];y=[0 100*sin(t)];
C=[112*cos(199.96*pi/180),112*sin(199.96*pi/180)];

A=[x(1),y(1)];B=[x(2),y(2)];D=[98*cos(40.1*pi/180),98*sin(40.1*pi/180)];
a=sqrt((D(1)-B(1))^2+(D(2)-B(2))^2);
b=sqrt((A(1)-D(1))^2+(A(2)-D(2))^2);
c=sqrt((A(1)-B(1))^2+(A(2)-B(2))^2);
cosC=(a^2+b^2-c^2)/(2*a*b);

```

```

d=40*pi/180
a=sqrt((C(1)-B(1))^2+(C(2)-B(2))^2);
b=sqrt((A(1)-C(1))^2+(A(2)-C(2))^2);
c=sqrt((A(1)-B(1))^2+(A(2)-B(2))^2);
c1=c;
cosC=(a^2+b^2-c^2)/(2*a*b);
aa1=acos(cosC)
a=sqrt((C(1)-D(1))^2+(C(2)-D(2))^2);
b=sqrt((A(1)-C(1))^2+(A(2)-C(2))^2);
c=sqrt((A(1)-D(1))^2+(A(2)-D(2))^2);
c2=c;
cosC=(a^2+b^2-c^2)/(2*a*b);
aa2=acos(cosC)
a=sqrt((C(1)-D(1))^2+(C(2)-D(2))^2);
b=sqrt((B(1)-C(1))^2+(B(2)-C(2))^2);
c=sqrt((B(1)-D(1))^2+(B(2)-D(2))^2);
c3=c;
cosC=(a^2+b^2-c^2)/(2*a*b);
aa3=acos(cosC);
m=[aa1,aa2,aa3];
cc=[c1,c2,c3];
a3=max(m);
c3=cc(find(m==a3));
xxx=find(m==a3);
[ad i]=setdiff(m,a3);
cc(xxx)=[];
m=m(sort(i))
a1=m(1);a2=m(2);
c1=100;c2=68;c3=100;
syms x;
f=sin(pi-a2-d+x)/sin(pi-a1-x)-sin(a2)/sin(a1);
p0=0;
tol=0.0000001;
maxK=1000;
[p,k,Y]=NTM(f,p0,tol,maxK);
fprintf("迭代值如下: ");
disp(Y);
x=Y(end)
y=d-x

```

## 附录 6

介绍： 第一题第（3）小问第二次迭代代码

%点2、3、4、5特殊情况

```
C=[76.9623 63.1241];
```

```
%C=[13.2557 96.5947];
```

```
%C=[-50.126 83.2211];
```

```
%C=[-92.8944 33.535];
```

```
A=[0 0];B=[100 0];D=[-93.4266,-33.9651];
```

```

a=sqrt((D(1)-B(1))^2+(D(2)-B(2))^2);
b=sqrt((A(1)-D(1))^2+(A(2)-D(2))^2);
c=sqrt((A(1)-B(1))^2+(A(2)-B(2))^2);
cosC=(a^2+b^2-c^2)/(2*a*b);
d=200*pi/180
a=sqrt((C(1)-B(1))^2+(C(2)-B(2))^2);
b=sqrt((A(1)-C(1))^2+(A(2)-C(2))^2);
c=sqrt((A(1)-B(1))^2+(A(2)-B(2))^2);
c1=c;
cosC=(a^2+b^2-c^2)/(2*a*b);
aa1=acos(cosC)
a=sqrt((C(1)-D(1))^2+(C(2)-D(2))^2);
b=sqrt((A(1)-C(1))^2+(A(2)-C(2))^2);
c=sqrt((A(1)-D(1))^2+(A(2)-D(2))^2);
c2=c;
cosC=(a^2+b^2-c^2)/(2*a*b);
aa2=acos(cosC)
a=sqrt((C(1)-D(1))^2+(C(2)-D(2))^2);
b=sqrt((B(1)-C(1))^2+(B(2)-C(2))^2);
c=sqrt((B(1)-D(1))^2+(B(2)-D(2))^2);
c3=c;
cosC=(a^2+b^2-c^2)/(2*a*b);
aa3=acos(cosC);
m=[aa1,aa2,aa3];
cc=[c1,c2,c3];
a3=max(m);
c3=cc(find(m==a3));
xxx=find(m==a3);
[ad i]=setdiff(m,a3);
cc(xxx)=[];
m=m(sort(i))
a1=m(1);a2=m(2);
c1=100;c2=100;c3=196.9616;
syms x;
f=sin(a2)/sin(a1)-sin(pi-d+x-a2)/sin(pi-x-a1);
tol=0.0000001;
maxK=1000;
[p,k,Y]=NTM(f,p0,tol,maxK);
fprintf("迭代值如下: ");
disp(Y);
x=Y(end)
y=d-x

%7、8、9特殊情况
C=[-51.3673 -88.7565];
C=[20.8285 -101.0852];
C=[80.9321 -61.3664];

A=[0 0];B=[100 0];D=[-93.4266,-33.9651];

```

```

a=sqrt((D(1)-B(1))^2+(D(2)-B(2))^2);
b=sqrt((A(1)-D(1))^2+(A(2)-D(2))^2);
c=sqrt((A(1)-B(1))^2+(A(2)-B(2))^2);
cosC=(a^2+b^2-c^2)/(2*a*b);
d=160*pi/180
a=sqrt((C(1)-B(1))^2+(C(2)-B(2))^2);
b=sqrt((A(1)-C(1))^2+(A(2)-C(2))^2);
c=sqrt((A(1)-B(1))^2+(A(2)-B(2))^2);
c1=c;
cosC=(a^2+b^2-c^2)/(2*a*b);
aa1=acos(cosC)
a=sqrt((C(1)-D(1))^2+(C(2)-D(2))^2);
b=sqrt((A(1)-C(1))^2+(A(2)-C(2))^2);
c=sqrt((A(1)-D(1))^2+(A(2)-D(2))^2);
c2=c;
cosC=(a^2+b^2-c^2)/(2*a*b);
aa2=acos(cosC)
a=sqrt((C(1)-D(1))^2+(C(2)-D(2))^2);
b=sqrt((B(1)-C(1))^2+(B(2)-C(2))^2);
c=sqrt((B(1)-D(1))^2+(B(2)-D(2))^2);
c3=c;
cosC=(a^2+b^2-c^2)/(2*a*b);
aa3=acos(cosC);
m=[aa1,aa2,aa3];
cc=[c1,c2,c3];
a3=max(m);
c3=cc(find(m==a3));
xxx=find(m==a3);
[ad i]=setdiff(m,a3);
cc(xxx)=[];
m=m(sort(i))
a1=m(1);a2=m(2);
c1=100;c2=100;c3=196.9616
syms x;
f=sin(a2)/sin(a1)-sin(pi-d+x-a2)/sin(pi-x-a1);
p0=0;
tol=0.0000001;
maxK=1000;
[p,k,Y]=NTM(f,p0,tol,maxK);
fprintf("迭代值如下: ");
disp(Y);
x=Y(end)
y=d-x

function [p,k,Y]=NTM(f,p0,tol,maxK)
%p0表示迭代初始值
%f表示要求解的方程
%maxK表示规定的最大迭代次数
%tolr表示允许误差

```

```

%k表示最终迭代的次数
%p表示最终迭代的值
syms x;
P(1)=p0;
k=2;
df=diff(f); %利用diff()函数计算f(x)的导数

P(k)=P(k-1)-subs(f,x,P(k-1))/subs(df,x,P(k-1)); %第二次迭代的结果
while k<=maxK
    err=abs(P(k)-P(k-1)); %err表示相邻的迭代值的差值
    if(err<tol)
        fprintf('迭代%d次即可满足允许误差值退出\n',k-1);
        break;
    end
    k=k+1;
    P(k)=P(k-1)-subs(f,x,P(k-1))/subs(df,x,P(k-1)); %迭代
end %共迭代了k-1次
if(k-1==maxK)
    disp("超过最大迭代次数！");
end
p=P(k);
k=k-1;
Y=P;
end

```

## 附录 7

### 介绍： 第一题第（3）小问第三次迭代代码

```

%点6、7特殊情况
C=[-93.4266 -33.9651];
C=[-50.0054 -85.9862];
A=[0 0];B=[100 0];D=[17.1455 98.3329];
a=sqrt((D(1)-B(1))^2+(D(2)-B(2))^2);
b=sqrt((A(1)-D(1))^2+(A(2)-D(2))^2);
c=sqrt((A(1)-B(1))^2+(A(2)-B(2))^2);
cosC=(a^2+b^2-c^2)/(2*a*b);
d=-80*pi/180
a=sqrt((C(1)-B(1))^2+(C(2)-B(2))^2);
b=sqrt((A(1)-C(1))^2+(A(2)-C(2))^2);
c=sqrt((A(1)-B(1))^2+(A(2)-B(2))^2);
c1=c;
cosC=(a^2+b^2-c^2)/(2*a*b);
aa1=acos(cosC)
a=sqrt((C(1)-D(1))^2+(C(2)-D(2))^2);
b=sqrt((A(1)-C(1))^2+(A(2)-C(2))^2);
c=sqrt((A(1)-D(1))^2+(A(2)-D(2))^2);
c2=c;
cosC=(a^2+b^2-c^2)/(2*a*b);
aa2=acos(cosC)

```



```

a=sqrt((C(1)-D(1))^2+(C(2)-D(2))^2);
b=sqrt((B(1)-C(1))^2+(B(2)-C(2))^2);
c=sqrt((B(1)-D(1))^2+(B(2)-D(2))^2)
c3=c;
cosC=(a^2+b^2-c^2)/(2*a*b);
aa3=acos(cosC);
m=[aa1,aa2,aa3];
cc=[c1,c2,c3];
a3=max(m);
c3=cc(find(m==a3));
xxx=find(m==a3);
[ad i]=setdiff(m,a3);
cc(xxx)=[];
m=m(sort(i))
a1=m(1);a2=m(2);
c1=100;c2=100;c3=128.5575;
syms x;
f=sin(a2)/sin(a1)-sin(pi-x-a2)/sin(pi-d+x-a1);
%f=sin(a1)/sin(a2)-sin(pi-x-a1)/sin(pi-d+x-a2);具体情况调用不同函数
tol=0.0000001;
maxK=1000;
[p,k,Y]=NTM(f,p0,tol,maxK);
fprintf("迭代值如下: ");
disp(Y);
x=Y(end)
y=d-x
%点2特殊情况
C=[76.5466 64.313];
A=[0 0];B=[100 0];D=[17.1455 98.3329];
a=sqrt((D(1)-B(1))^2+(D(2)-B(2))^2);
b=sqrt((A(1)-D(1))^2+(A(2)-D(2))^2);
c=sqrt((A(1)-B(1))^2+(A(2)-B(2))^2);
cosC=(a^2+b^2-c^2)/(2*a*b);
d=80*pi/180
a=sqrt((C(1)-B(1))^2+(C(2)-B(2))^2);
b=sqrt((A(1)-C(1))^2+(A(2)-C(2))^2);
c=sqrt((A(1)-B(1))^2+(A(2)-B(2))^2);
c1=c;
cosC=(a^2+b^2-c^2)/(2*a*b);
aa1=acos(cosC)
a=sqrt((C(1)-D(1))^2+(C(2)-D(2))^2);
b=sqrt((A(1)-C(1))^2+(A(2)-C(2))^2);
c=sqrt((A(1)-D(1))^2+(A(2)-D(2))^2);
c2=c;
cosC=(a^2+b^2-c^2)/(2*a*b);
aa2=acos(cosC)
a=sqrt((C(1)-D(1))^2+(C(2)-D(2))^2);
b=sqrt((B(1)-C(1))^2+(B(2)-C(2))^2);
c=sqrt((B(1)-D(1))^2+(B(2)-D(2))^2)
c3=c;

```

```

cosC=(a^2+b^2-c^2)/(2*a*b);
aa3=acos(cosC);
m=[aa1,aa2,aa3];
cc=[c1,c2,c3];
a3=max(m);
c3=cc(find(m==a3));
xxx=find(m==a3);
[ad i]=setdiff(m,a3);
cc(xxx)=[];
m=m(sort(i))
a1=m(1);a2=m(2);
c1=100;c2=100;c3=128.5575;
syms x;
f=sin(a2)/sin(a1)-sin(pi-x-a2)/sin(pi-d+x-a1);
%f=sin(a1)/sin(a2)-sin(pi-x-a1)/sin(pi-d+x-a2);具体情况调用不同函数
tol=0.0000001;
maxK=1000;
[p,k,Y]=NTM(f,p0,tol,maxK);
fprintf("迭代值如下: ");
disp(Y);
x=Y(end)
y=d-x
%点4、5特殊情况
C=[-49.9856 86.1775];
C=[-93.4933 33.9298];
A=[0 0];B=[100 0];D=[17.1455 98.3329];
a=sqrt((D(1)-B(1))^2+(D(2)-B(2))^2);
b=sqrt((A(1)-D(1))^2+(A(2)-D(2))^2);
c=sqrt((A(1)-B(1))^2+(A(2)-B(2))^2);
cosC=(a^2+b^2-c^2)/(2*a*b);
d=50*pi/180
a=sqrt((C(1)-B(1))^2+(C(2)-B(2))^2);
b=sqrt((A(1)-C(1))^2+(A(2)-C(2))^2);
c=sqrt((A(1)-B(1))^2+(A(2)-B(2))^2);
c1=c;
cosC=(a^2+b^2-c^2)/(2*a*b);
aa1=acos(cosC)
a=sqrt((C(1)-D(1))^2+(C(2)-D(2))^2);
b=sqrt((A(1)-C(1))^2+(A(2)-C(2))^2);
c=sqrt((A(1)-D(1))^2+(A(2)-D(2))^2);
c2=c;
cosC=(a^2+b^2-c^2)/(2*a*b);
aa2=acos(cosC)
a=sqrt((C(1)-D(1))^2+(C(2)-D(2))^2);
b=sqrt((B(1)-C(1))^2+(B(2)-C(2))^2);
c=sqrt((B(1)-D(1))^2+(B(2)-D(2))^2)
c3=c;
cosC=(a^2+b^2-c^2)/(2*a*b);
aa3=acos(cosC);
m=[aa1,aa2,aa3];

```

```

cc=[c1,c2,c3];
a3=max(m);
c3=cc(find(m==a3));
xxx=find(m==a3);
[ad i]=setdiff(m,a3);
cc(xxx)=[];
m=m(sort(i))
a1=m(1);a2=m(2);
c1=100;c2=100;c3=128.5575;
syms x;
f=100*sin(a2)/(128.5575*sin(a1))-sin(pi-d+x-a2)/sin(pi-x-a1);
%f=sin(a1)/sin(a2)-sin(pi-x-a1)/sin(pi-d+x-a2);具体情况调用不同函数
tol=0.0000001;
maxK=1000;
[p,k,Y]=NTM(f,p0,tol,maxK);
fprintf("迭代值如下: ");
disp(Y);
x=Y(end)
y=d-x
%点8、9特殊情况
C=[15.799 -92.9819];
C=[76.3405 -64.4687];
A=[0 0];B=[100 0];D=[17.1455 98.3329];
a=sqrt((D(1)-B(1))^2+(D(2)-B(2))^2);
b=sqrt((A(1)-D(1))^2+(A(2)-D(2))^2);
c=sqrt((A(1)-B(1))^2+(A(2)-B(2))^2);
cosC=(a^2+b^2-c^2)/(2*a*b);
d=50*pi/180
a=sqrt((C(1)-B(1))^2+(C(2)-B(2))^2);
b=sqrt((A(1)-C(1))^2+(A(2)-C(2))^2);
c=sqrt((A(1)-B(1))^2+(A(2)-B(2))^2);
c1=c;
cosC=(a^2+b^2-c^2)/(2*a*b);
aa1=acos(cosC)
a=sqrt((C(1)-D(1))^2+(C(2)-D(2))^2);
b=sqrt((A(1)-C(1))^2+(A(2)-C(2))^2);
c=sqrt((A(1)-D(1))^2+(A(2)-D(2))^2);
c2=c;
cosC=(a^2+b^2-c^2)/(2*a*b);
aa2=acos(cosC)
a=sqrt((C(1)-D(1))^2+(C(2)-D(2))^2);
b=sqrt((B(1)-C(1))^2+(B(2)-C(2))^2);
c=sqrt((B(1)-D(1))^2+(B(2)-D(2))^2)
c3=c;
cosC=(a^2+b^2-c^2)/(2*a*b);
aa3=acos(cosC);
m=[aa1,aa2,aa3];
cc=[c1,c2,c3];
a3=max(m);
c3=cc(find(m==a3));

```

```

xxx=find(m==a3);
[ad i]=setdiff(m,a3);
cc(xxx)=[];
m=m(sort(i))
a1=m(1);a2=m(2);
c1=100;c2=100;c3=128.5575;
syms x;
f=sin(a2)/sin(a1)-sin(pi-d+x-a2)/sin(pi-x-a1);
tol=0.0000001;
maxK=1000;
[p,k,Y]=NTM(f,p0,tol,maxK);
fprintf("迭代值如下: ");
disp(Y);
x=Y(end)
y=d-x
function [p,k,Y]=NTM(f,p0,tol,maxK)
%p0表示迭代初始值
%f表示要求解的方程
%maxK表示规定的最大迭代次数
%tolr表示允许误差
%k表示最终迭代的次数
%p表示最终迭代的值
syms x;
P(1)=p0;
k=2;
df=diff(f); %利用diff()函数计算f(x)的导数

P(k)=P(k-1)-subs(f,x,P(k-1))/subs(df,x,P(k-1)); %第二次迭代的结果
while k<=maxK
    err=abs(P(k)-P(k-1)); %err表示相邻的迭代值的差值
    if(err<tol)
        fprintf('迭代%d次即可满足允许误差值退出\n',k-1);
        break;
    end
    k=k+1;
    P(k)=P(k-1)-subs(f,x,P(k-1))/subs(df,x,P(k-1)); %迭代
end %共迭代了k-1次
if(k-1==maxK)
    disp("超过最大迭代次数!");
end
p=P(k);
k=k-1;
Y=P;
end

```

## 附录 8

介绍：第一题第（3）小问计算误差代码（以第一次迭代为例）

```
a1=[76.6044 17.3648 -50 -93.9693 -93.9693 -50 17.3648 76.6044
```

```

64.2788 98.4808 86.6025 34.202 -34.202 -86.6025 -98.4808 -64.2788];
a2=[76.9623 13.2557 -50.126 -92.8944 -93.4266 -51.3673 20.8285 80.9321
63.1241 96.5947 83.2211 33.535 -33.9651 -88.7565 -101.0852 -61.3664];
m=a1-a2
l=[]
for i=1:8
    l=[l sqrt(m(1,i)^2+m(2,i)^2)]
end

```

#### 附录 9

介绍： 第一题第（3）问有偏差位置移向理想位置计算函数

```

%(x0,y0)是理想位置的坐标
%(x1, y1) 是假的实际位置
%(x2, y2) 是实际位置
%(x3,y3) 是迭代出的下一次的坐标
function [x3,y3]=iter(x0,y0,x1,y1,x2,y2)
x3=x2+x0-x1;
y3=y2+y0-y1;
end

```

#### 附录 10

介绍： 第二题与稳健性分析作图代码

```

%第二题
%%
a=[0,0
    25*sqrt(3),25
    50*sqrt(3),50
    75*sqrt(3),75
    100*sqrt(3),100
    0,50
    25*sqrt(3),75
    50*sqrt(3),100
    75*sqrt(3),125
    0,100
    25*sqrt(3),125
    50*sqrt(3),150
    0,150
    25*sqrt(3),175
    0,200
    ]
scatter(a(:,1),a(:,2),'b*')
hold on
for i=1:max(size(a(:,1)))
    c = num2str(i-1);
    text(a(i,1),a(i,2),c);
end

```

```

a=[0,0
    25*sqrt(3),25
    50*sqrt(3),50
    75*sqrt(3),75
    100*sqrt(3),100
    0,50
    75*sqrt(3),125
    0,100
    50*sqrt(3),150
    0,150
    25*sqrt(3),175
    0,200];
b=[25*sqrt(3),75
    50*sqrt(3),100
    25*sqrt(3),125];
scatter(a(:,1),a(:,2),'b*')
hold on
scatter(b(:,1),b(:,2),'r*')
hold on
x=25*sqrt(3);y=75;r=50;
rectangle('Position', [x-r,y-r,2*r,2*r], 'Curvature', [1 1], 'EdgeColor', 'r');
hold on
x=50*sqrt(3);y=100;r=50;
rectangle('Position', [x-r,y-r,2*r,2*r], 'Curvature', [1 1], 'EdgeColor', 'r');
hold on
x=25*sqrt(3);y=125;r=50;
rectangle('Position', [x-r,y-r,2*r,2*r], 'Curvature', [1 1], 'EdgeColor', 'r');
%%
a=[0,0
    25*sqrt(3),25
    50*sqrt(3),50
    75*sqrt(3),75
    100*sqrt(3),100
    0,50
    75*sqrt(3),125
    0,100
    50*sqrt(3),150
    0,150
    25*sqrt(3),175
    0,200];
b=[25*sqrt(3),75
    50*sqrt(3),100
    25*sqrt(3),125];
scatter(a(:,1),a(:,2),'b*')
hold on
scatter(b(:,1),b(:,2),'r*')
hold on
x=25*sqrt(3);y=75;r=50;
rectangle('Position', [x-r,y-r,2*r,2*r], 'Curvature', [1 1], 'EdgeColor', 'r');
hold on

```

```

x=50*sqrt(3);y=100;r=50;
rectangle('Position', [x-r,y-r,2*r,2*r], 'Curvature', [1 1], 'EdgeColor', 'r');
hold on
x=25*sqrt(3);y=125;r=50;
rectangle('Position', [x-r,y-r,2*r,2*r], 'Curvature', [1 1], 'EdgeColor', 'r');
%%
a=[0,0
    50*sqrt(3),50
    50*sqrt(3),150
    0,200
    0,100
    100*sqrt(3),100];
b=[0,150
    25*sqrt(3),175
    25*sqrt(3),125
    0,50
    25*sqrt(3),25
    25*sqrt(3),75
    75*sqrt(3),75
    75*sqrt(3),125
    50*sqrt(3),100

    ];
scatter(a(:,1),a(:,2),'b*')
hold on
scatter(b(:,1),b(:,2),'r*')
hold on
x=0;y=150;r=50;
rectangle('Position', [x-r,y-r,2*r,2*r], 'Curvature', [1 1], 'EdgeColor', 'r');
hold on
x=0;y=50;r=50;
rectangle('Position', [x-r,y-r,2*r,2*r], 'Curvature', [1 1], 'EdgeColor', 'r');
hold on
x=75*sqrt(3);y=75;r=50;
rectangle('Position', [x-r,y-r,2*r,2*r], 'Curvature', [1 1], 'EdgeColor', 'r');
%%
a=[0,0
    25*sqrt(3),25
    50*sqrt(3),50
    75*sqrt(3),75
    100*sqrt(3),100
    0,50
    75*sqrt(3),125
    0,100
    50*sqrt(3),150
    0,150
    25*sqrt(3),175
    0,200
    -25*sqrt(3),-25
    -25*sqrt(3),25

```

```

-25*sqrt(3),75
-25*sqrt(3),125
-25*sqrt(3),175
-25*sqrt(3),225
]
b=[50*sqrt(3),100
25*sqrt(3),75
25*sqrt(3),125
];
scatter(a(:,1),a(:,2),'b*')
hold on
scatter(b(:,1),b(:,2),'r*')
hold on

%稳健性分析可视化
box=[0.573 1.1459 1.7189 2.2918 2.8648 3.4377 4.0107 4.5837 5.1566 5.7296
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 2 2];
scatter(box(1,:),box(2:4,:),'k*')
hold on
scatter(box(1,:),box(3,:), 'kv')
hold on
scatter(box(1,:),box(4,:), 'ko')
hold on

```