

# 无人机纯方位无源定位模型

## 摘要

无人机编队飞行的用途相当广泛，它不仅给人以视觉上的享受，还可用于军事，实现密集突防。本文主要研究纯方位无源定位情形下的无人机定位问题，建立了接收信号无人机的定位模型，利用几何分析法、优化模型进行求解，并对模型进行验证。

**针对问题 1.1：**无人机的定位问题。定位模型建立过程如下：首先，基于 10 架无人机各自的位置信息和编号信息，以及接收信号无人机所收到的方位角度信息，由夹角余弦定理建立利用定位三角形的重心确定方程组的解；其次，通过方位角度信息与发射信号无人机编号之间的对应关系，建立 6 组关联角度和编号的方程，得到 6 组解，并将所求解与理想解之间最小值的平方和设置为目标函数，从而找到与理想解最近的解；最后，确定定位模型最终解并得出定位模型，利用问题 1.3 中的数据对模型进行检验，效果良好。

**针对问题 1.2：**发射源位置位置时无无人机的定位问题。由问题 1.1 知只需要由接收信号确定发射源即可。通过对角度规律的分析，在偏离位置较小的情况下，只需额外添加 2 架发射信号无人机就可以有效定位。

**针对问题 1.3：**圆形编队调整位置的方案选择问题。首先，基于 2 架发射信号无人机的编号已知且位置无偏差的情形，选取两个固定点（FY00、FY01）和一到两个未知点（与理想位置最近的无人机）发射信号；其次，基于定位模型得到的位置和理想位置之间的误差，对位置的调整进行迭代处理；以误差平方和为目标函数建立优化模型，利用 *Matlab* 求解得出需进行 6 次调整才能使 9 架无人机均匀分布于圆周上，最终距离误差在 0.01 米以内。

**针对问题 2：**利用锥形编队的几何特征，将 15 架无人机用 6 个圆形编队进行覆盖。选取编号是 FY03、FY05、FY07、FY08、FY09 和 FY14 的无人机作为 6 个圆心，依次调整 6 个圆周上无人机的位置使之能够均匀分布。基于问题 1.3 求解的方法，对锥形中 15 架无人机的位置进行调整，设计出锥形编队调整方案的算法。

最后，我们对模型进行了讨论和分析，做出了客观的优缺点评价。

**关键词：**纯方位无源定位    定位三角性    目标优化    余弦公式

## 一、问题重述

### 1.1 问题背景

在遂行编队飞行中，无人机有两种类型：一种用于发射信号，另一种用于接收信号。每一架无人机之间通过信号的传递微调方位，即利用纯方位无源定位对无人机的位置进行调整，从而保持编队的队形。其中，被动接收信号的无人机将收到“自身与任何两架发射信号的无人机连线间夹角”的方向信息。编队中每一架无人机都有固定的编号。

### 1.2 需要解决的问题

1 架编号为 FY00 的无人机位于圆心，其余 9 架编号为 FY01 ~ FY09 的无人机则均匀分布于某一圆周上，形成圆形编队。飞行中，所有无人机的高度是相同且不变的。

**问题 1.1:** 已知发射信号无人机（位于圆心的 FY00 和编队中另外 2 架）的编号且其位置无偏差，建立被动接收信号无人机（位置略有偏差）的定位模型。

**问题 1.2:** 接收信号无人机位置略有偏差，假设发射信号无人机（FY00、FY01 和编队中若干编号未知的）位置无偏差，求出为有效定位无人机所需添加的发射信号无人机的数量。

**问题 1.3:** 要求形成圆形编队，圆心有 1 架无人机，半径为  $100m$  的圆周上均匀分布着另外 9 架无人机。根据表 1 的数据，已知部分无人机的初始位置略有偏差，给出无人机位置具体的调整方案（不考虑每次调整的时间），使 9 架无人机最后在某个圆周上均匀分布。每次调整中，最多有 4 架发射信号的无人机（FY00 和圆周上至多 3 架），其余均为接收方向信息的无人机。

**问题 2:** 应用纯方位无源定位的方法，设计无人机位置的调整方案，使得直线上相邻两架无人机的距离是相等的，即形成锥形编队。

## 二、问题分析

问题 1 为无人机在飞行形成圆形编队过程中的定位与位置调整方案的选择问题。其中，问题 1.1、问题 1.2 均为定位问题，问题 1.3 为方案选择问题。问题 2 为无人机集群在形成锥形编队过程中的位置调整方案的选择问题。

## 2.1 问题 1.1 的分析

对于问题 1.1, 3 架发射信号无人机的编号已知且位置无偏差, 因此被动接收信号的无人机所收到的方位信息即为实际的位置信息。由于接收信号无人机的位置有偏差, 考虑到其将收到 3 个夹角的角度信息且夹角与 3 架已知编号无人机的对应关系是未知的, 因此本文先通过几何关系分析并建立夹角角度与已知编号无人机位置的关系式, 再从有限解中找到与被动定位无人机的理想位置最近的解, 以期建立三站纯方位无源定位模型。

## 2.2 问题 1.2 的分析

对于问题 1.2, 接收信号无人机的位置稍有偏差且其编号是否已知并未明确, 因此将其分成编号已知和编号未知两种情况分析。此问采用角度大小与相对位置关系对未知的编号进行确定, 巧妙地将问题转化为“使编号确定的最少的无人机数目”这一目标求解问题。根据圆周角等于其所对圆心角的一半, 分析得出圆周上两架无人机与接收源所形成夹角的角度为  $20n^\circ$  ( $n$  为整数), 以及根据已确定编号无人机的数量得到各个夹角的角度关系式, 尝试建立与发射信号无人机数量和夹角角度相关的有效定位模型。

## 2.3 问题 1.3 的分析

问题 1.3 需要将初始位置有偏差的无人机调整到理想位置上, 考虑到编号为 FY00、FY01 的无人机的初始位置是无偏差的, 且发射信号无人机的数量是有限制的, 因此该问选取“两个定点、一个或两个未知点”发射信号, 对接收信号的无人机进行定位并给出调整方案。其次, 考虑实际定位与理想位置之间的误差, 此问需通过迭代多次调整位置, 以建立定位模型和设计调整方案的选取算法。

## 2.4 问题 2 的分析

问题 2 是问题 1.3 的拓展。此问基于问题 1.3 的圆形编队调整位置的算法, 将分别以 FY05、FY08、FY09 为圆心, 靠近圆心最近的 6 架无人机分别均匀分布于 3 个圆周上, 依次调整 3 个圆周上的无人机位置, 建立定位模型和设计调整方案的选取算法。

### 三、模型的假设

1. 假设接收信号无人机的编号和理想位置已知；
2. 假设无人机之间传递的所有信息均是准确无误的；
3. 所有无人机在飞行过程中的高度是相同且不变的；
4. 调整无人机的方位时，不考虑每次调整的时间。

### 四、符号说明

符号	意义
$i, j (i, j = 0, 1, 2, \dots, 9)$	发射信号无人机的编号
$k (k = 0, 1, 2, \dots, 9)$	接收信号无人机的编号
$\alpha_{ij} (i, j = 0, 1, 2, \dots, 9)$	FY0i 与 FY0j 发射信号之间的夹角
$(x_k, y_k) (k = 0, 1, 2, \dots, 9)$	FY0k 的坐标
$(x_m^{(*)}, y_m^{(*)}) (k = 0, 1, 2, \dots, 9)$	编号为 $m$ 的无人机的理想坐标
$(x_m^{(q)}, y_m^{(q)}) (m = 3, 4, 5, \dots, 9)$	编号为 $m$ 的无人机接收信号时定位到的坐标
$(x_m^{(k)}, y_m^{(k)}) (m = 3, 4, 5, \dots, 9)$	第 $k$ 次迭代求出的编号为 $m$ 的无人机的坐标
$d_m$	编号为 $m$ 的无人机定位位置与实际位置间的距离
$d$	编号为 $m$ 的无人机定位位置与实际位置间的平均距离
$d_m^k$	第 $k$ 次迭代求出的编号为 $m$ 的无人机与理想位置的距离
$d^k$	第 $k$ 次迭代求出的所有无人机与理想位置的最大距离
$E^2$	能够容忍的最大距离平方

### 五、模型建立、求解与检验

#### 5.1 问题 1.1 的模型建立与求解

以编号为 FY00 的无人机所在位置为坐标原点，指向 FY01 的方向为  $x$  轴正方向建立坐标系。

假设发射源无人机编号为  $0, i, j$ ，接收信号无人机编号为  $k$ ，其接收到的方位信息为  $(\alpha_1, \alpha_2, \alpha_3)$ 。由于不知道方位信息与发射源编号之间的对应关系，因此共有  $A_3^2$  种情形。而问题 1.1 中已知无人机稍有偏差，利用夹角余弦公式计算对应的坐标信息，并对六种结果选取离理想坐标最近的作为定位坐标。

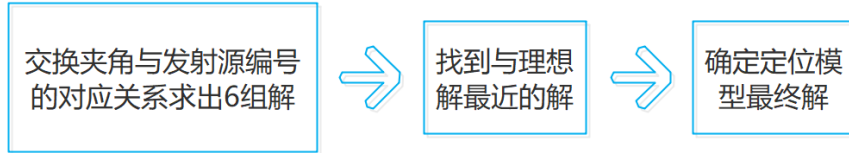


图 1: 思路图

### 5.1.1 信号几何关系分析

下面我们选择编号为  $0, i, j$  作为信号发射源，选择  $k$  作为信号接收源进行绘图分析，示意图如下：

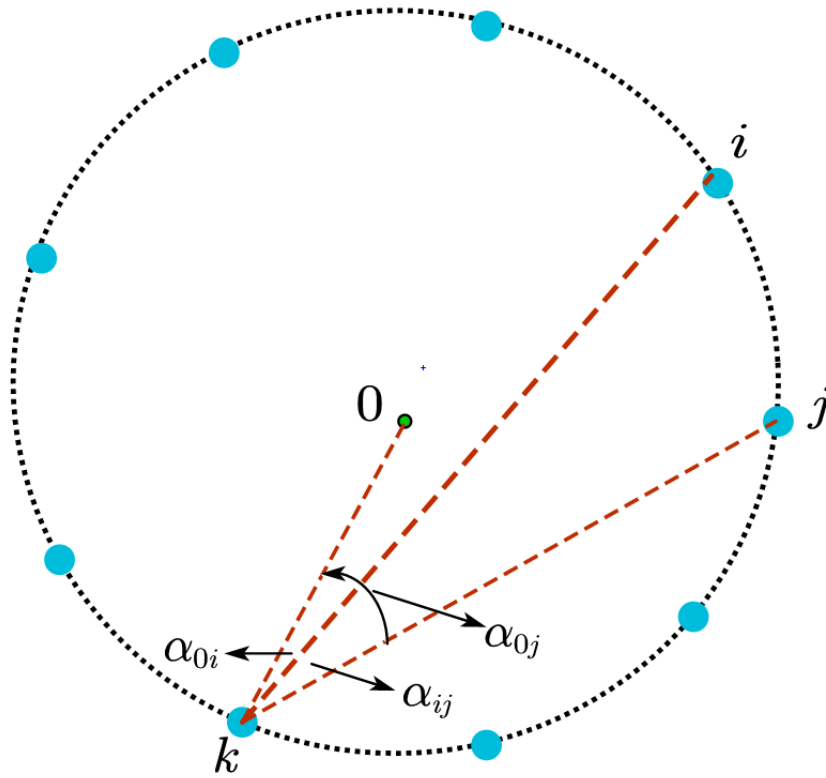


图 2: 信号几何关系图

通过分析几何关系，由夹角余弦公式  $\cos(\vec{m}, \vec{n}) = \frac{\vec{m} \cdot \vec{n}}{|\vec{m}| \cdot |\vec{n}|}$  可得下列关系式：

$$\begin{cases} \cos(\alpha_{0i}) = \frac{(x_0-x)(x_i-x)+(y_0-y)(y_i-y)}{\sqrt{(x_0-x)^2+(y_0-y)^2}\sqrt{(x_i-x)^2+(y_i-y)^2}} \\ \cos(\alpha_{0j}) = \frac{(x_0-x)(x_j-x)+(y_0-y)(y_j-y)}{\sqrt{(x_0-x)^2+(y_0-y)^2}\sqrt{(x_j-x)^2+(y_j-y)^2}} \\ \cos(\alpha_{ij}) = \frac{(x_i-x)(x_j-x)+(y_i-y)(y_j-y)}{\sqrt{(x_i-x)^2+(y_i-y)^2}\sqrt{(x_j-x)^2+(y_j-y)^2}} \end{cases} \quad (1)$$

其中， $\alpha_{0i}$  表示编号为 0 和  $i$  的无人机发射的信号之间的夹角， $\alpha_{0j}$  表示 0 与  $j$  对应夹角， $\alpha_{ij}$  表示  $i$  与  $j$  对应夹角， $(x_0, y_0), (x_i, y_i), (x_j, y_j)$  分别表示编号为 0、 $i$ 、 $j$  的无人机的坐标。

### 5.1.2 编号与夹角对应关系已知情况下方程组的求解

考虑在编号与夹角对应关系已知的情形下， $\alpha_{0i}, \alpha_{0j}, \alpha_{ij}, (x_0, y_0), (x_i, y_i), (x_j, y_j)$  已知，共计三个方程、两个未知量  $P(x, y)$ 。

#### 5.1.2.1 解的存在性分析：

(1) 理论上只需要任意两个方程即可解出  $P$  点坐标，但是由于方程组 1 中任意方程均超过一次，可能产生冗余解。由于已知  $P$  点的理想位置，可以通过解与理想位置的距离排除掉明显不合理的情况。

(2) 现在通过三个方程求解  $P$  点坐标：理想情况下， $P$  点坐标满足方程组 1，方程组 1 解的个数为有限个，一定可以通过求解方程组 1 求出  $P$  点坐标；实际上，由数值误差，每两个方程求出的解（总计三组）可能存在一定的误差，使得求解三个方程无法成功。

#### 5.1.2.2 方程组解的处理：

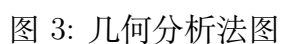
接收信号无人机收到了三个方向发来的信号，即三条始发于已知编号无人机的射线，每两条射线有一个夹角。

(1) 理想情况下，三线交于一点；

(2) 存在较小数值误差的情况下，三线交于三点且形成了一个较小的三角性区域，定位坐标大致位于三角性区域内。

使用几何图形说明这一关系，图中最大的蓝色点表示接收信号无人机，红色实线无误差时的情况；蓝色虚线表示有误差时的情况，红色实心点表示有误差时，由三个方程中两个方程求出的解，图中三条射线围成了一个三角形区域  $ABC$ ，可以通过这三个点围成的区域估计出更合理的解。

示意图如下：



将三角形  $ABC$  称为定位三角形<sup>[2]</sup>, 问题转化为求“定位三角形”的质心。

$$\begin{cases} x = \frac{x_A + x_B + x_C}{3} \\ y = \frac{y_A + y_B + y_C}{3} \end{cases} \quad (2)$$
$$\begin{cases} x = \frac{ax_A + bx_B + cx_C}{a+b+c} \\ y = \frac{ay_A + by_B + cy_C}{a+b+c} \end{cases} \quad (3)$$

从中选择重心法代入模型求解。

### 5.1.3 编号与夹角对应关系未知情况下方程组的求解

已知： $\{\alpha | \alpha = \alpha_{0i}, \alpha_{0j}, \alpha_{ij}\} = \{\alpha | \alpha = \alpha_1, \alpha_2, \alpha_3\}$  通过下列任一条件与方程组 1 联立求得结果。

$$\begin{cases} \alpha_{0i} = \alpha_1 \\ \alpha_{0j} = \alpha_2 \\ \alpha_{ij} = \alpha_3 \end{cases} \begin{cases} \alpha_{0i} = \alpha_1 \\ \alpha_{0j} = \alpha_3 \\ \alpha_{ij} = \alpha_2 \end{cases} \begin{cases} \alpha_{0i} = \alpha_2 \\ \alpha_{0j} = \alpha_1 \\ \alpha_{ij} = \alpha_3 \end{cases} \begin{cases} \alpha_{0i} = \alpha_2 \\ \alpha_{0j} = \alpha_3 \\ \alpha_{ij} = \alpha_1 \end{cases} \begin{cases} \alpha_{0i} = \alpha_3 \\ \alpha_{0j} = \alpha_1 \\ \alpha_{ij} = \alpha_2 \end{cases} \begin{cases} \alpha_{0i} = \alpha_3 \\ \alpha_{0j} = \alpha_2 \\ \alpha_{ij} = \alpha_1 \end{cases} \quad (4)$$

求得 6 组解： $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), (x^{(3)}, y^{(3)}), (x^{(4)}, y^{(4)}), (x^{(5)}, y^{(5)}), (x^{(6)}, y^{(6)})$ ，由接收信号的无人机编号可得，对应的理想解为  $(x^{(*)}, y^{(*)})$ 。则由

$$\begin{aligned} \text{Min} \quad & (x^{(k)} - x^{(*)})^2 + (y^{(k)} - y^{(*)})^2 \\ \text{s.t.} \quad & k \in \{1, 2, 3, 4, 5, 6\} \end{aligned} \quad (5)$$

可得最终解为： $(x^{(k)}, y^{(k)})$ 。

### 5.1.4 方程组求解的一般步骤

最终求解按照以下规则：

- (1) 在方程 4 中六选一，给定夹角与已知编号一个对应关系；
  - (2) 通过方程组 1 中每两个方程得到一个解，若有冗余解，按照与理想解的距离远近排除；
  - (3) 将得到的三个解选取重心法进行平均，得到偏差较小的最优解；
  - (4) 此时得到了一个解  $(x^{(1)}, y^{(1)})$ ，重新进行第 1 步不重复的交换顺序，求出所有基本解  $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), (x^{(3)}, y^{(3)}), (x^{(4)}, y^{(4)}), (x^{(5)}, y^{(5)}), (x^{(6)}, y^{(6)})$ ；
  - (5) 根据与理想解  $(x^{(*)}, y^{(*)})$  的距离求出最终解  $(x^{(k)}, y^{(k)})$ ；
  - (6) 定位完成。
- 求解流程图，如下图 4 显示：



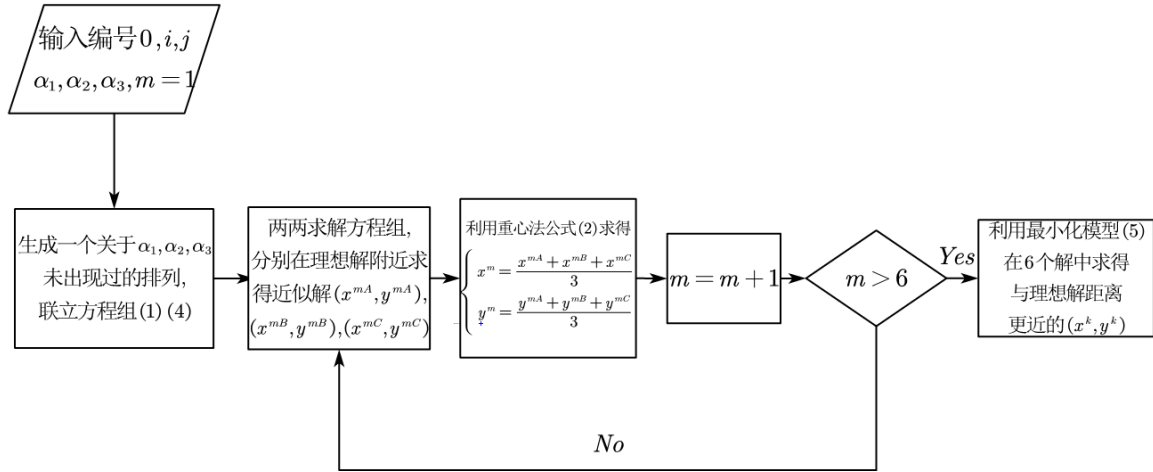


图 4: 1.1 流程图

### 5.1.5 模型求解与检验

#### 5.1.5.1 模型误差检验的方法<sup>[1]</sup>:

距离:

$$d_m = \sqrt{(x_m^{(k)} - x_m^{(q)})^2 + (y_m^{(k)} - y_m^{(q)})^2} \quad (6)$$

平均距离:

$$d = \frac{\sqrt{(x_m^{(k)} - x_m^{(q)})^2 + (y_m^{(k)} - y_m^{(q)})^2}}{7} \quad (7)$$

$x$  坐标的相对误差:

$$e_m^{(x)} = \left| \frac{x_m^{(k)} - x_m^{(q)}}{x_m^{(q)}} \right| \quad (8)$$

$y$  坐标的相对误差:

$$e_m^{(y)} = \left| \frac{y_m^{(k)} - y_m^{(q)}}{y_m^{(q)}} \right| \quad (9)$$

坐标相对误差:

$$\bar{e}_m = \frac{e_m^{(x)} + e_m^{(y)}}{2} \quad (10)$$

平均坐标相对误差:

$$\bar{e} = \frac{\sum_{i=3}^9 \bar{e}_i}{7} \quad (11)$$

其中  $m = 3, 4, 5, 6, 7, 8, 9$ , 表示接收信号的无人机编号;  $x_m^{(q)}$  表示编号为  $m$  的无人机接收信号时定位到的横坐标,  $y_m^{(q)}$  此时定位到的纵坐标;  $x_m^{(k)}$  表示编号为  $m$  的无人机

实际的横坐标,  $y_m^{(k)}$  此时实际的纵坐标。

#### 5.1.5.2 下面进行模型的求解和检验:

选择发射信号的无人机编号为 FY00、FY01 和 FY02 验证模型的可行性, 自定义满足题目要求的数据加随机小误差, 使用建立的模型和 *Matlab* 函数 *vpasolve*, 以理想解为初始值, 并以发射源到接收源实际的方位信息为输入信息, 依次算出 FY03、FY04、FY05、FY06、FY07、FY08 和 FY09 的定位坐标, 分别计算与实际坐标的距离、 $x$  坐标的相对误差、 $y$  坐标的相对误差, 并求出平均坐标距离、坐标的平均相对误差, 从而绘制误差条形图并检验模型的优劣。

通过对比各点的理想坐标和各点的定位坐标, 发现定位坐标十分精确。且如图 6, FY00、FY01、FY02 为发射源模拟求解的坐标与实际坐标的距离条形图及坐标的相对偏差条形图显示: 求解模型可解, 误差数量级分别在  $10^{-13}$ ,  $10^{-14}$ , 说明精确度很高。

x 的理想值	y 的理想值	x 的计算值	y 的计算值
19.04420117	110.3690102	19.04420117	110.3690102
-52.10273289	91.16087552	-52.10273289	91.16087552
-92.00770215	33.74289178	-92.00770215	33.74289178
-105.2722907	-38.23277145	-105.2722907	-38.23277145
-52.38886564	-90.99674037	-52.38886564	-90.99674037
17.30380046	-96.46024305	17.30380046	-96.46024305

表 1: 理想坐标与数据坐标

x 的相对误差	y 的相对误差	坐标相对误差	距离
4.10412E-15	1.28758E-16	2.11644E-15	7.94411E-14
8.1824E-16	9.35326E-16	8.76783E-16	9.53293E-14
3.08906E-16	2.10576E-15	1.20733E-15	7.65278E-14
2.29485E-15	5.2037E-15	3.74928E-15	3.12962E-13
5.42514E-16	3.12338E-16	4.27426E-16	4.01944E-14
9.64976E-15	1.17859E-15	5.41417E-15	2.02005E-13

表 2: 模型误差

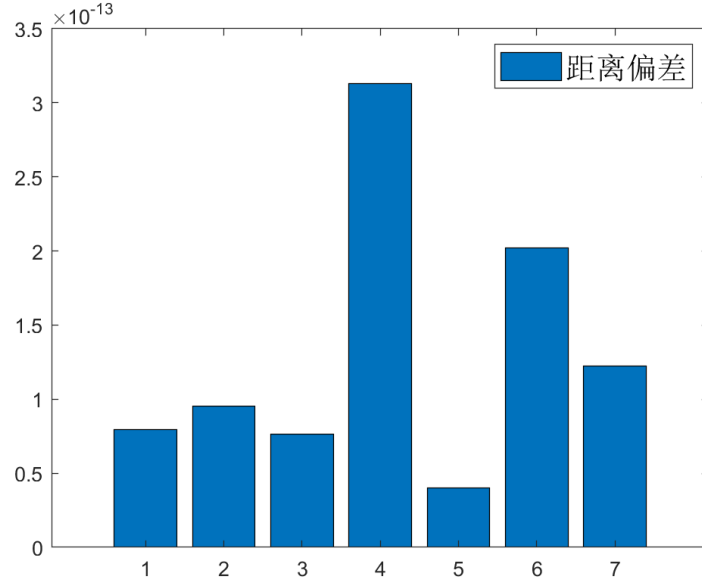


图 5: 距离条形图

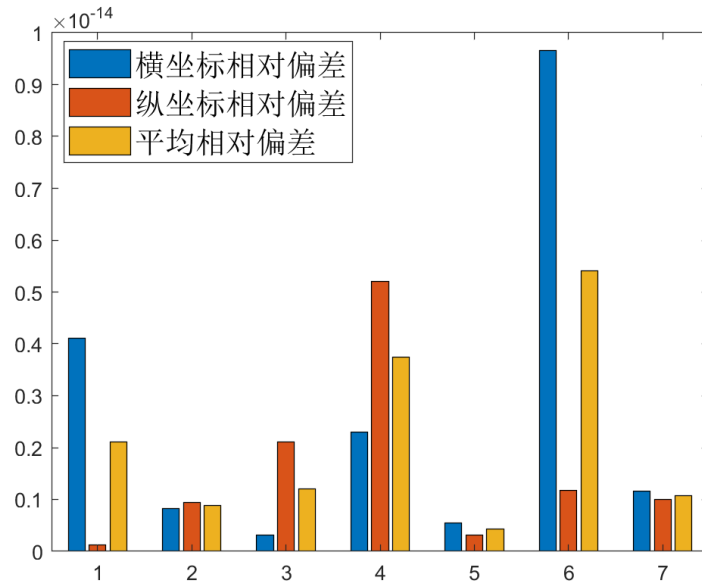


图 6: 相对误差条形图

## 5.2 问题 1.2 的模型建立与求解

继续假设  $P$  号无人机被动接收信号，其位置满足稍有偏差，FY00、FY01 及另外若干编号未知的无人机发射信号，其位置无偏差。

当编号已知时，问题转化为问题 1.1，即除了 FY00、FY01 之外，只需要额外一架无人机即可实现定位。当编号未知时，如果能通过角度的大小和相对关系确定具体的编

号，问题即可解决。所以，问题 (1.2) 转化为使发射信号源编号确定所需的最少的无人机数目。

### 5.2.1 均匀分布和圆心角、圆周角规律分析

#### 5.2.1.1 飞行编队均匀分布的特性：

由于九架无人机均匀地分布在圆上，每两架无人机之间的距离相等，且相邻两架无人机到 FY00 号无人机之间的夹角为  $40^\circ$ ，任意两架无人机到 FY00 号无人机之间的夹角为  $40n^\circ$  ( $n$  为整数)。

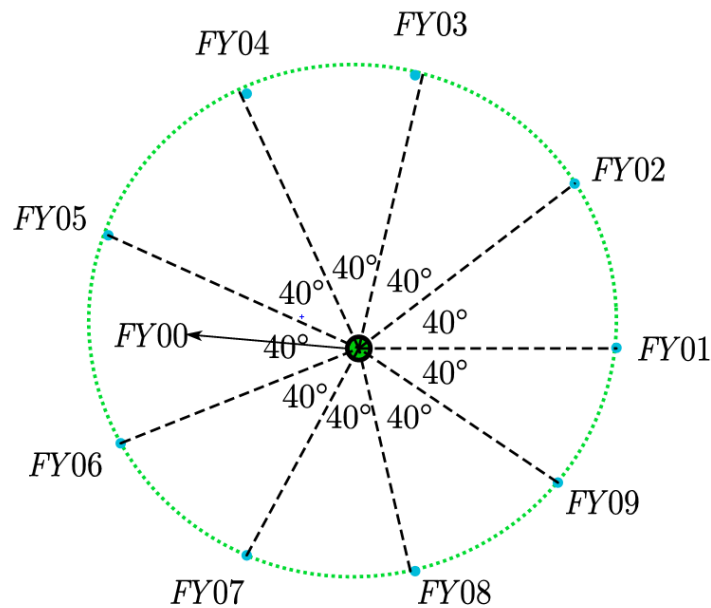


图 7: 均匀分布示意图

#### 5.2.1.2 两个无人机在圆周上时接收到夹角的角度规律：

当两个在圆周上的无人机向接收信号的无人机发射信号时，夹角构成圆周角。根据圆周角为圆心角一半的特性，且圆心角均为  $40n^\circ$  ( $n$  为整数)，可知此类夹角一定是  $20n^\circ$  ( $n$  为整数)。

#### 5.2.1.3 无人机 FY00 与任意编号无人机接收到夹角的角度规律：

当无人机 FY00 与任意编号无人机向接收信号的无人机发射信号时，夹角构成圆周角。在下文图 8 中分析其对应的圆心角的角度规律：

记接收源为点  $P$ ，FY00 为点  $O$ ，FY01 为点  $A$ ，另外发射源在点为  $B$ 。连结点  $P$  与原点  $O$ ，延长并与圆相交于点  $Q$ 。在图示特殊情况下，研究出 FY00 与另一个圆心上的无人机 (这里是 FY02) 对应的夹角， $\angle OPB$  为圆周角，对应圆心角为  $\angle QOB$ ，可以

看出圆心角  $\angle QOB$  为  $20^\circ$  的倍数且不是  $40^\circ$  的倍数，则圆周角  $\angle OPB$  为  $10^\circ$  的倍数且不是  $20^\circ$  的倍数。

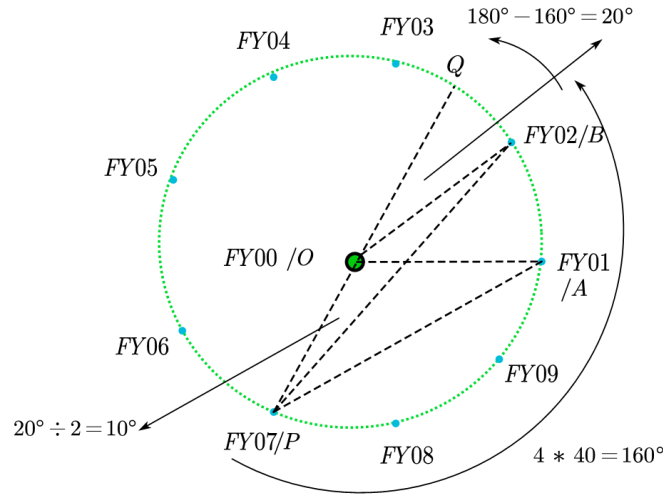


图 8: 角度规律分析图

在更一般的情形下，圆心角  $\angle QOB = 180^\circ - \angle POA = 180^\circ - 40n^\circ = 40m^\circ + 20^\circ$ ，则圆周角  $\angle OPB = 20m^\circ + 10^\circ$ 。可以得出结论：无人机 FY00 与任意圆上无人机到接收源构成的夹角满足规律  $20m^\circ + 10^\circ$ 。

其中， $m, n$  均为整数。

#### 5.2.1.4 角度规律总结：

经过分析，角度规律可以总结为：圆周上两无人机与接收源形成角  $20n^\circ$  ( $n$  为整数)；圆周上一无人机、无人机 FY00 与接收源形成角  $20m^\circ + 10^\circ$  ( $m, n$  为整数)

根据题意：编号为 FY00、FY01 的无人机及若干编号未知的无人机发射信号。当需要确定编号的额外无人机为一架时，三个角分别为  $20n^\circ$ 、 $20m_1^\circ + 10^\circ$ 、 $20m_2^\circ + 10^\circ$ ；当需要确定编号的额外无人机为两架时，六个角分别为  $20n_1^\circ$ 、 $20n_2^\circ$ 、 $20n_3^\circ$ 、 $20m_1^\circ + 10^\circ$ 、 $20m_2^\circ + 10^\circ$ 、 $20m_3^\circ + 10^\circ \dots$

其中， $m_i, n_i$  均为整数。

#### 5.2.2 添加一个额外发射源确定无人机编号

下面考虑仅通过添加一个额外发射源确定各无人机编号的方法。根据规律分析的结论，已知三个角的角度形式为： $20n^\circ$ 、 $20m_1^\circ + 10^\circ$ 、 $20m_2^\circ + 10^\circ$  ( $m_1, m_2, n$  均为整数)。

将此时的接收无人机记为  $k$ ，发射信号的无人机编号为  $0, 1, i$ ，记无人机  $01, 0i, 1i$  对应的夹角  $\alpha_{01}, \alpha_{0i}, \alpha_{1i}$ ，接收到三个夹角为  $\alpha_1, \alpha_2, \alpha_3$ 。

### 5.2.2.1 分析 $\alpha_{01}$ :

$\alpha_{01}$  为无人机发射源 FY00、FY01 及接收信号的无人机  $k$  形成的夹角。接收信号的无人机知晓自己的编号  $k$ ，由编号  $k, 0, 1$  可以求出理想情况下夹角  $\alpha_{01}^*$  的值。

求解方法如图 9 所示：

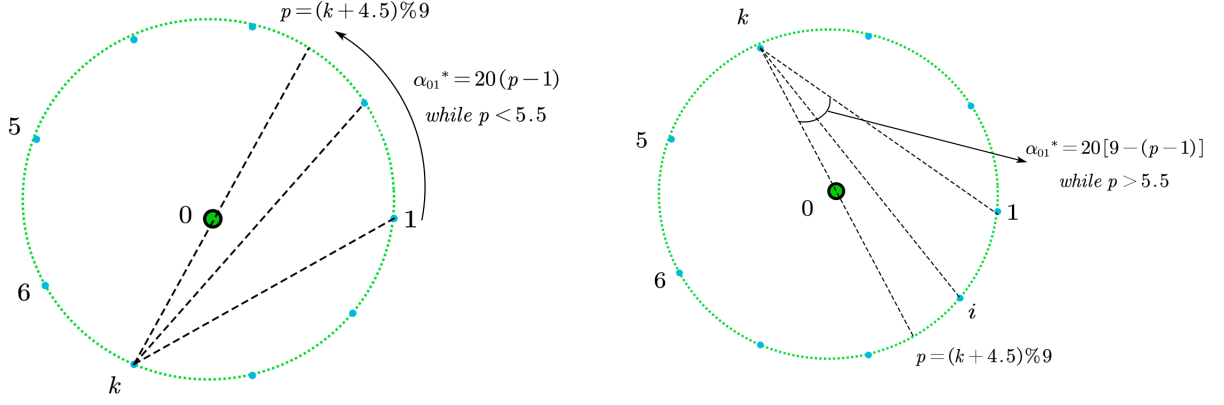


图 9:  $\alpha_{01}^*$  的求解

延长接收源与圆心的连线交圆于某点  $Q$ 。记  $p = (k + 4.5) \% 9$  为点  $Q$  的虚拟编号，虚拟编号的值表示点  $Q$  在编号  $[p]$  对应点至编号  $[p] + 1$  对应点相连圆弧的中间位置。编号夹角  $\alpha_{01}^*$  的值转化为编号  $p$ 、编号 1 与接收源编号  $k$  的点形成的夹角。

此时存在两种情况：

- (1) 当编号  $p < 5.5$  时， $20(p - 1)$  对应劣弧的度数，满足条件；
- (2) 当编号  $p > 5.5$  时， $20(p - 1)$  对应优弧的度数，需要转化为劣弧的度数，才能与接收的信息吻合，此时  $\alpha_{01}^* = 20(9 - (p - 1))$ 。

于是，求得  $\alpha_{01}$  的理想解  $\alpha_{01}^*$ 。下面根据理想解匹配得到 FY00、FY01 与接收点之间形成的夹角  $\alpha_{01}$ 。

$$\begin{aligned} \text{Min } S_1 &= |\alpha_{01}^* - \alpha_{k_1}| \\ \text{s.t. } \begin{cases} k_1 \in \{1, 2, 3\} \\ |\alpha_{01}^* - \alpha_{k_1}| < \text{Maxerror} \end{cases} \end{aligned} \quad (12)$$

其中， $k_1$  表示  $\alpha_{k_1} = \alpha_{01}$ ，即接收到的夹角信息， $\alpha_{k_1}^*$  表示当接收点位于理想位置时应该接受到的夹角信息， $\text{Maxerror}$  表示允许的误差限。

于是匹配可以得到  $\alpha_1$  的实际解  $\alpha_{k_1}$ ，也可得到  $\alpha_1, \alpha_2, \alpha_3$  与  $\alpha_{01}, \alpha_{0i}, \alpha_{1i}$  中的其中一对对应关系，只需再确定一对对应关系及求出  $i$ ，即可求解。

### 5.2.2.2 求解编号 $i$ :

- (1) 根据角度形式明确所有角的对应关系：

根据角度规律的分析,  $\alpha_{1i} = 20n$  ( $n$  为整数), 现有剩余的两个角形式分别为  $20n^\circ, 20m^\circ + 10^\circ$ , 则可以确定  $\alpha_{1i}$  的值。

$$\begin{aligned} \text{Min } S_2 &= |\alpha_{k_2} - 20n| \\ \text{s.t. } &\begin{cases} k_2 \in \{1, 2, 3\} \\ k_2 \neq k_1 \\ |\alpha_{k_2} - 20n| < \text{Maxerror} \end{cases} \end{aligned} \quad (13)$$

其中,  $k_2$  表示  $\alpha_{k_2} = \alpha_{1i}$ , 即接收到  $1, i$  对应的夹角信息,  $\text{Maxerror}$  表示允许的误差限。

此时找到了第二对对应关系, 只需求出  $i$  即可明确任意信号发射源的编号及任意两个发射源所对应的夹角。

(1) 根据  $1, i$  夹角得到  $i$  的两个解  $i_1, i_2$ :

对  $1, i$  对应的夹角进行分析, 即可得到如下关系式:

$$\begin{cases} \alpha_{k_2} = 20(k_2 - 1) & k_2 = 2, 3, 4, 5 \\ \alpha_{k_2} = 20(9 - (k_2 - 1)) & k_2 = 6, 7, 8, 9 \end{cases} \quad (14)$$

(2) 根据  $0, i$  夹角得到  $i$  的两个解  $i_3, i_4$ :

分析  $0, i$  对应夹角, 类似于分析  $0, 1$  对应的夹角, 得到如下关系式:

$$\begin{cases} \alpha_{k_3} = 20(p - i) & k_3 = 2, 3, 4, 5 \\ \alpha_{k_3} = 20(9 - (p - i)) & k_3 = 6, 7, 8, 9 \end{cases} \quad (15)$$

其中,  $k_3$  表示  $\alpha_{k_3} = \alpha_{0i}$ , 即接收到的  $0, i$  对应夹角信息,  $p = (k + 4.5)\%9$ , 表示虚拟编号  $Q$  的值 (见前文图 9 所示)。

(3) 最终确定编号  $i$ : 显然, 真实的  $i$  值在集合  $\{i_1, i_2\}$  与集合  $\{i_3, i_4\}$  的交中一定存在; 但是如果存在:  $i_1 = i_3, i_2 = i_4$  或者  $i_1 = i_4, i_2 = i_3$  的情况, 即存在两组解, 依然无法确定最终解。

下图 10 说明了这种情况不可能发生:

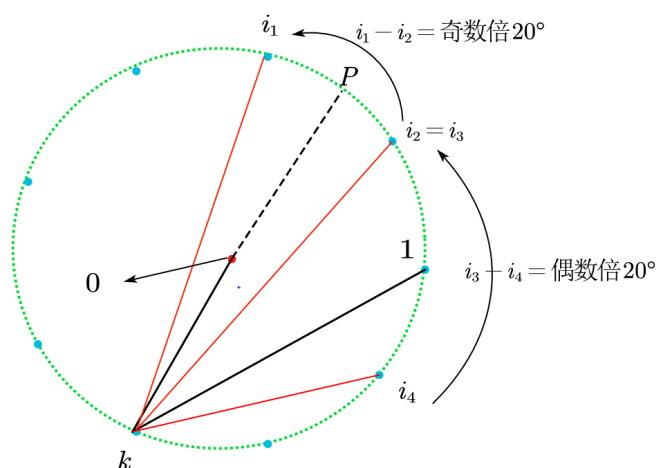


图 10:  $i$  的唯一性分析图

$i_1$  和  $i_2$  关于线段  $OP$  对称, 则对应圆心角为奇数倍个  $20^\circ$ ;  $i_3$  和  $i_4$  关于 0 和 1 位置的连线对称, 则对应圆心角为偶数倍个  $20^\circ$ ; 当  $i_2 = i_3$  时,  $i_1$  一定不等于  $i_4$ , 即不存在两个解。最终解为:  $i = \{i_1, i_2\} \cup \{i_3, i_4\}$ 。

#### 5.2.2.3 结论:

发射源 FY00、FY01 和圆周上任意一架无人机发射信号, 可以通过分析角度关系求出未知编号无人机的编号, 并得到三个角与分别与两个发射源的对应关系。

#### 5.2.3 模型求解与检验

设计验证模式: 已知编号 FY00、FY01 和编号  $i$  (令  $i = 2$  进行验证) 的无人机发射信号, 自定义满足条件的数据, 使得接受信号无人机位置稍有偏差, 依次选择其他无人机验证求解编号  $i$  的值, 与给定  $i$  的值进行比较验证。求解结果见表格??, 结果显示上述求解未知编号的方法可行、正确。

表 3: 求解编号验证表

发射	接收	$\alpha_1$	对应发点	$\alpha_2$	对应发点	$\alpha_3$	对应发点	逆推编号
0,1,2	3	45.88249	9, 0	56.72687	9, 1	10.84437	0, 1	2
	4	38.85423	9, 1	25.55319	9, 0	13.30104	0, 1	2
	5	32.2266	2, 1	14.11228	0, 1	18.11432	2, 0	2
	6	25.8194	2, 1	11.71116	0, 1	14.10823	2, 0	2
	7	22.4301	2, 1	8.44134	0, 1	13.98876	2, 0	2
	8	19.68821	2, 1	6.467809	0, 1	13.2204	2, 0	2
	9	17.12872	2, 1	6.050137	0, 1	11.07858	2, 0	2

注: Matlab 中用 1 表示编号为 FY00 的无人机, 编号 2 在 Matlab 中表示为 3。



## 5.3 问题 1.3 的模型建立与求解

### 5.3.1 模型的建立

无人机编队均匀分布在半径  $r = 100$  米的圆上或位于圆心，通过 FY00 与圆周上最多三架无人机发射信号，帮助编队中的其他无人机确定自身的位置。

分析可知：

(1) 理论上，除圆心外圆周上需要的无人机架数至少为两架；由题意，圆周上的无人机架数为两架或三架；

(2) 在初始位置数据中，满足位于理想位置的无人机为 FY00、FY01，共两架；

(3) 接收信号的无人机无法得知发射信号无人机的实际坐标，只能以发射信号无人机的理想坐标进行定位，于是产生了定位误差，需要通过迭代多次调整求得误差更小的解；

(4) 由于已知无人机编号及理想位置，可以求得每架无人机与理想位置的距离来衡量误差，下一轮可以选择误差最小的三或四个点进行定位。

(5) FY00、FY01 位置无误差，每次固定选择这两个点，再另外选择一个或两个距离与理想距离最小的无人机发射信号。

使用问题 1.1 中建立的模型进行求解，每次除 FY00、FY01 额外选取一个点误差最小的点进行定位。求解过程可以用流程图表示，见图 11：

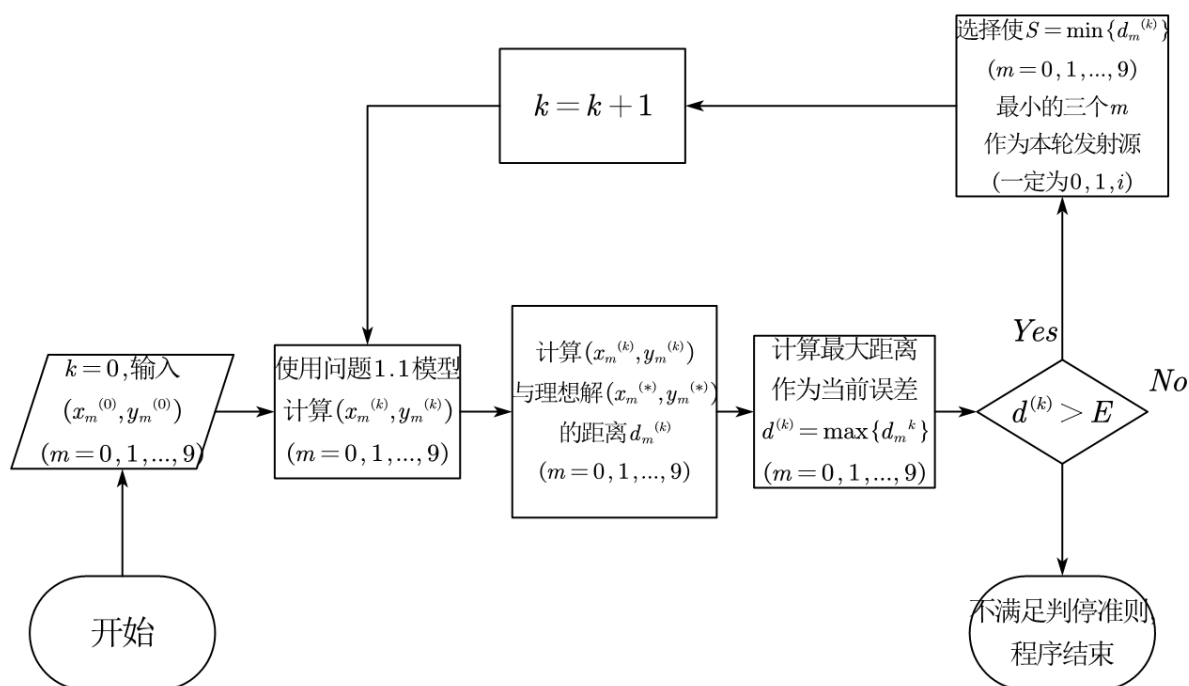


图 11: 模型求解流程图

第  $k$  轮迭代第  $m$  架无人机的误差:

$$d_m^k = \sqrt{(x_m^k - x_m^*)^2 + (y_m^k - y_m^*)^2} \quad m = 0, 1, 2, \dots, 9 \quad (16)$$

第  $k$  轮迭代的误差:

$$d^k = \text{Max}\{d_m^k\} \quad m = 0, 1, 2, \dots, 9 \quad (17)$$

其中,  $(x_m^{(k)}, y_m^{(k)})$  表示第  $k$  次迭代求出的编号为  $m$  的无人机的位置,  $(x_m^{(*)}, y_m^{(*)})$  表示编号为  $m$  的无人机的理想位置,  $d_m^k$  表示第  $k$  次迭代求出的编号为  $m$  的无人机与理想位置的距离,  $d^k$  表示第  $k$  次迭代求出的所有无人机与理想位置的最大距离, 能够容忍的最大距离平方为  $E$ 。

### 5.3.2 模型的求解

按照流程图所示的方法求解, 令容忍的最大距离平方为  $E^2 = 0.01$ , 可以通过五次迭代实现调整。五次迭代的误差下降图和无人机分布图, 分别为图 12 及图 13。

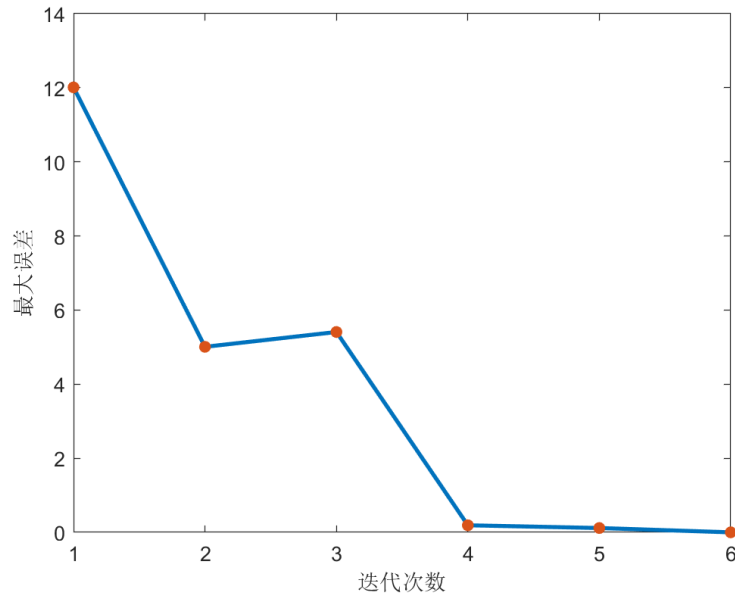


图 12: 误差折线图

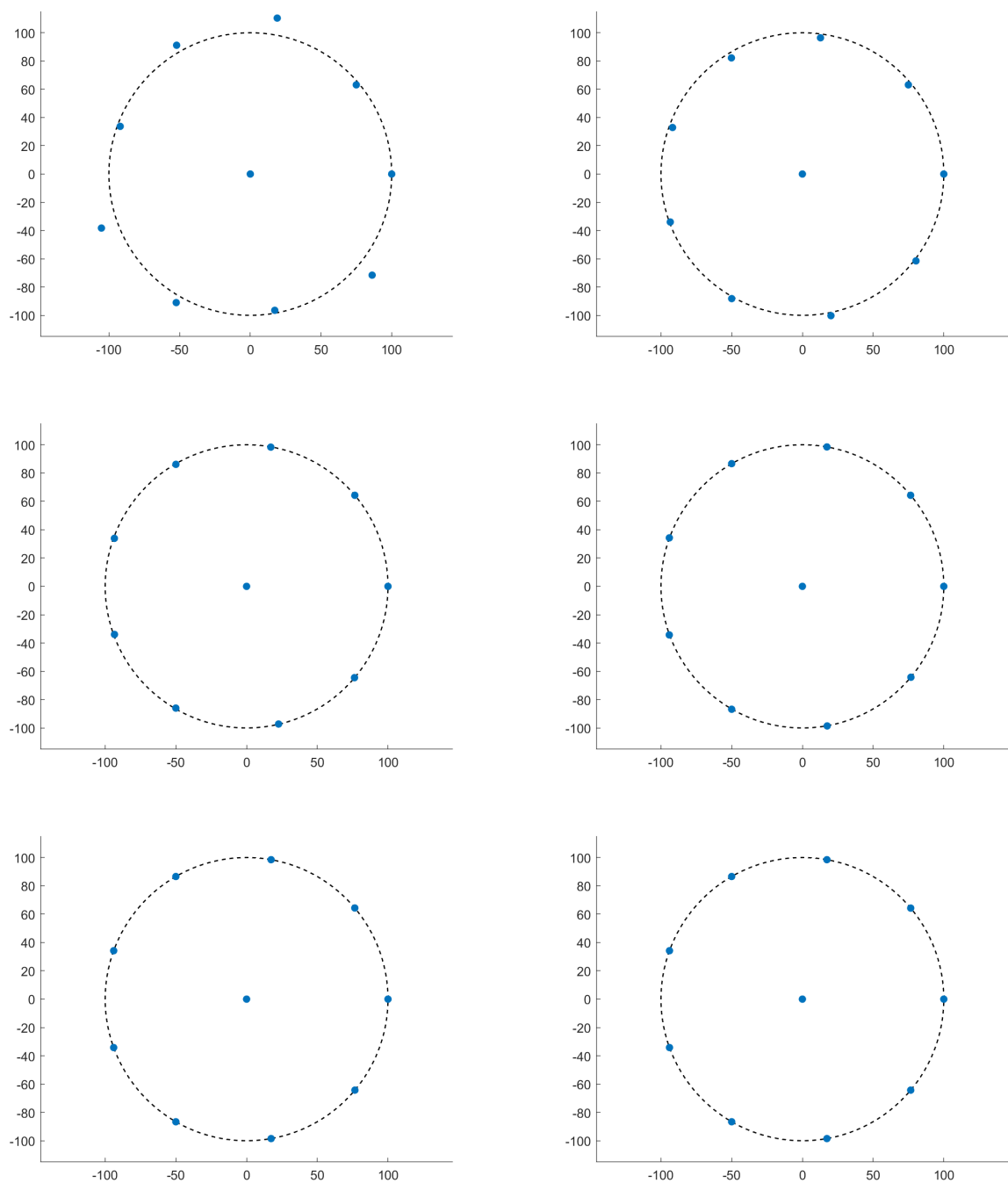


图 13: 迭代调整过程图

由图 13, 迭代过程的调整过程图可知, 在第一次和第二次迭代的结果显示了肉眼可见的误差; 而当到达第三次之后, 误差已经无法明显观察到接近于 0, 说明模型可行性及精度; 而观察误差折现图 13, 当到达第 4 次时, 误差已经接近于 0, 说明问题已经基本解决, 模型精度高。

## 5.4 问题 2 的模型建立与求解

### 5.5 模型建立

问题 2 考虑锥形编队无人机队形，直线上相邻两架无人机间距为  $50m$ 。

(1) 假设该编队中接收信号的无人机与问题 1 中情形相似，均与理想距离稍有偏差；

(2) 由问题 1.1 的分析可知，只需要一个圆心和两个该圆上的点即可对其他圆上的点进行定位；

(3) 由问题 1.3 的分析可知，当一个圆通过锥形编队中至少三个点时，能够知道圆心和圆上一个点的理想位置且其他位置稍有偏差时，可以通过问题 1.3 的模型能够将该圆上的所有点的坐标调整至理想位置；

(4) 该锥形编队，可以通过相互之间存在相交关系的若干圆进行覆盖；

问题转化为在锥形编队中构造圆形编队，向周围不同的圆发散求解，每次只有三架无人机发射信号，位于同一个圆上的其他无人机接收信号，确定部分无人机的位置，然后确定其他圆上的无人机的位置。求解顺序的演示图 14：

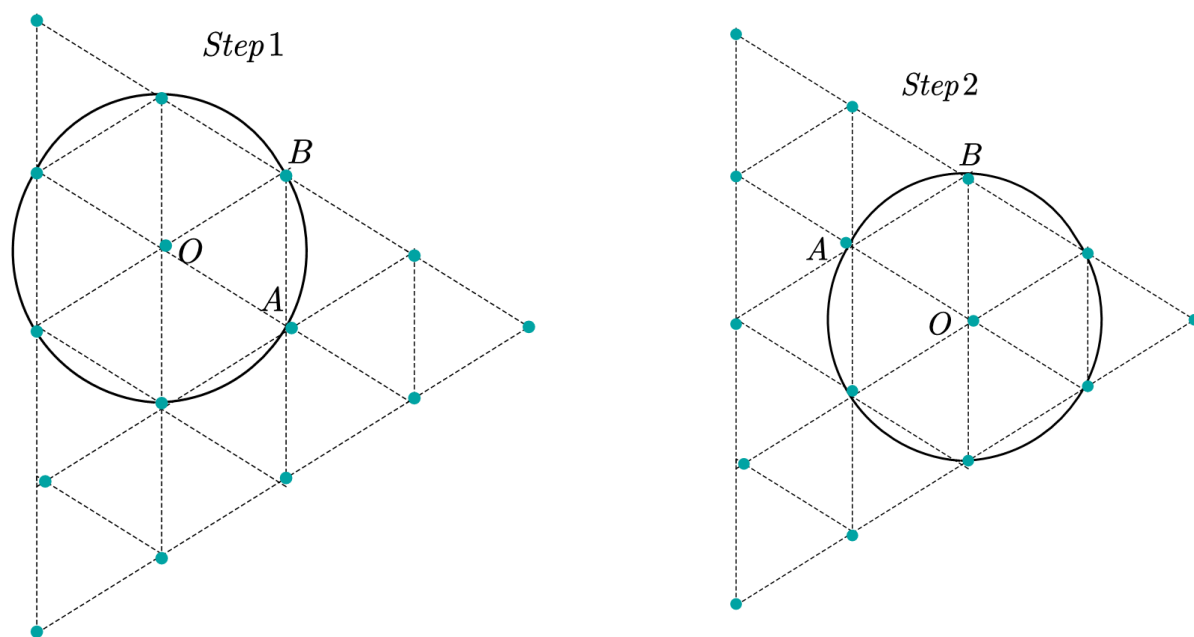


图 14: 问题 2 求解模型过程图 1-2

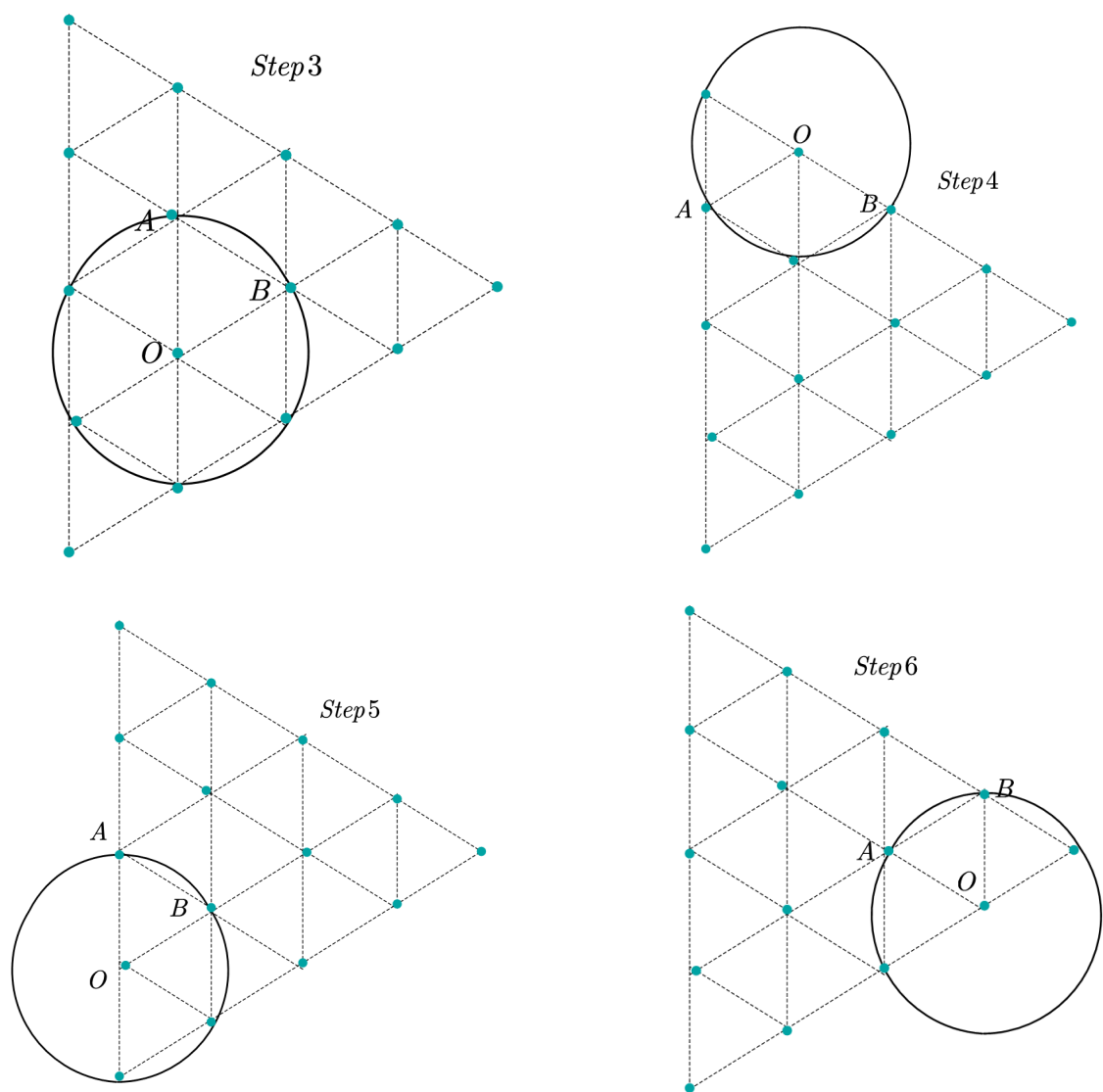


图 15: 问题 2 求解模型过程图 4-6

## 5.6 模型求解及检验

建系：这里假设直线上相邻两架无人机的间距相等为 200 米，以 1 为原点，1 到 5 的方向为正方向，初始精确点定位 Step1 图中的  $O, A$  点。

建立锥形定位模型，编号信息见图 16，检验的误差图见图 17。

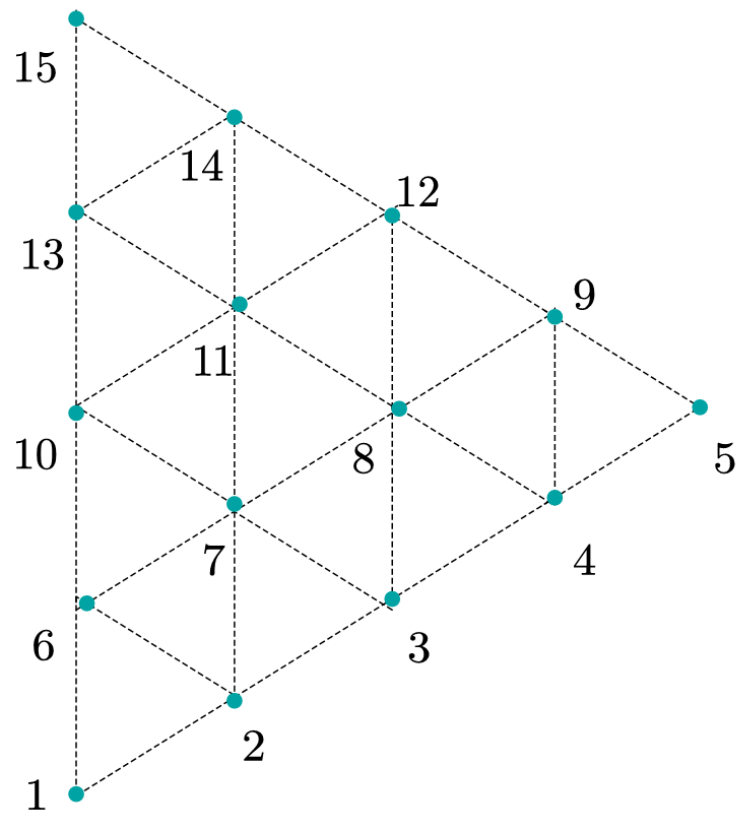


图 16: 锥形图

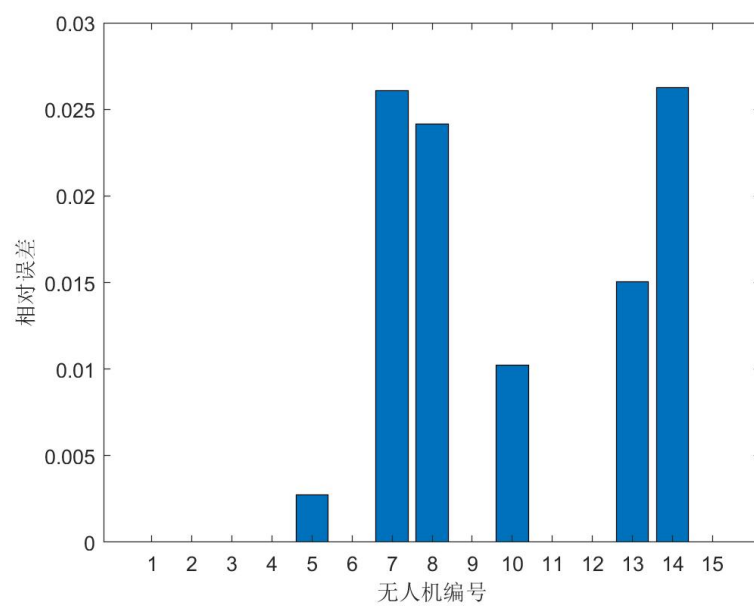


图 17: 问题 2 误差图

## 六、优缺点分析

### 6.1 优点

(1) 理想解的运用：通过分析实际解与理想解的距离，可以将求解限制在某一领域内。于是可以通过将理想解赋为初值，运用 *Matlab* 的 *vpasolve* 函数结合初值，快速求出解；

(2) 通过分析几何关系，将问题 1.2 转化为了问题 1.1，简化了模型的求解难度；

(3) 通过分析锥形的几何特征，利用六个圆覆盖锥形上的 15 个点，从而将该问题转化为问题 1.3。

### 6.2 缺点

夹角余弦方程组的求解有一定的不稳定性，给求解代码的书写带来了不小的难度。

## 参考文献

- [1] 洪胜孙. 基于方位角测量的无源定位算法研究. 硕士, 哈尔滨工程大学, 2010.
- [2] 杨巍. 基于贝叶斯估计的多站纯方位无源定位及优化, 2015.

## 附录 A 通用函数 1

"comp\_alpha.m"

```
1 % 代码段
2 function output = comp_alpha(x1y1,x2y2,x3y3,xy)
3 % 约定1-2之间的夹角是 $\alpha_1$ ,2-3之间的夹角是 $\alpha_2$ ,1-3之间的
   夹角是 $\alpha_3$ 
4 [inner_product1,mod1] = inner_product_model(x1y1,x2y2,xy);
5 [inner_product2,mod2] = inner_product_model(x2y2,x3y3,xy);
6 [inner_product3,mod3] = inner_product_model(x1y1,x3y3,xy);
7 alpha1 = acos(inner_product1/mod1);
8 alpha2 = acos(inner_product2/mod2);
9 alpha3 = acos(inner_product3/mod3);
10 output = [alpha1,alpha2,alpha3];
11 end
```

## 附录 B 通用函数 2

"inner\_product\_model.m"

```
1 % 代码段
2 function output = comp_alpha(x1y1,x2y2,x3y3,xy)
3 % 约定1-2之间的夹角是 $\alpha_1$ ,2-3之间的夹角是 $\alpha_2$ ,1-3之间的
   夹角是 $\alpha_3$ 
4 [inner_product1,mod1] = inner_product_model(x1y1,x2y2,xy);
5 [inner_product2,mod2] = inner_product_model(x2y2,x3y3,xy);
6 [inner_product3,mod3] = inner_product_model(x1y1,x3y3,xy);
7 alpha1 = acos(inner_product1/mod1);
8 alpha2 = acos(inner_product2/mod2);
9 alpha3 = acos(inner_product3/mod3);
10 output = [alpha1,alpha2,alpha3];
11 end
```



## 附录 C 问题 1.1 源代码 1

"shumo\_B\_1\_12.m"

```
1 function output = shumo_B_1_12(input1,input2,input3,input4,
    input5,input6,input7)
2 % 3个发送位置的坐标
3 FYsend1 = input1; % 圆心 直角坐标
4 FYsend2 = input2; % FY01
5 FYsend3 = input3; % FY02
6 FYsend = [FYsend1;FYsend2;FYsend3];
7
8 % 接收信息,只有三个角度的大小信息
9 alpha1 = input4; % 弧度
10 alpha2 = input5;
11 alpha3 = input6;
12
13 % 找到所有可能的情况
14 FYsend_perm = perms([1 2 3]); % 给出1 2 3这三个发射位置编号
    的排列
15 n = size(FYsend_perm,1);
16 error = +inf; % 设定所求出的位置和目标位置误差的初始值为正
    无穷
17 temp_xy = []; % 记录FYsend_perm中误差最小的情况对应位置
18
19 syms x y
20 ini = input7; % 设定初值为目标位置
21 for i = 1:n
22 temp_xy_code1 = FYsend_perm(i,[1 2]); % alpha1对应的两个发
    射点编号
23 temp_xy_1_1 = FYsend(temp_xy_code1(1),:);
24 temp_xy_1_2 = FYsend(temp_xy_code1(2),:);
25 [inner_product1,model1] = inner_product_model(temp_xy_1_1,
```

```

    temp_xy_1_2,[x y]);
26
27 temp_xy_code2 = FYsend_perm(i,[2 3]); % alpha2对应的两个发
    射点编号
28 temp_xy_2_1 = FYsend(temp_xy_code2(1),:);
29 temp_xy_2_2 = FYsend(temp_xy_code2(2),:);
30 [inner_product2,model2] = inner_product_model(temp_xy_2_1,
    temp_xy_2_2,[x y]);
31
32 temp_xy_code3 = FYsend_perm(i,[1 3]); % alpha3对应的两个发
    射点编号
33 temp_xy_3_1 = FYsend(temp_xy_code3(1),:);
34 temp_xy_3_2 = FYsend(temp_xy_code3(2),:);
35 [inner_product3,model3] = inner_product_model(temp_xy_3_1,
    temp_xy_3_2,[x y]);
36 % 定位
37 egns = [cos(alpha1) - inner_product1/model1 == 0;
38 cos(alpha2) - inner_product2/model2 == 0;
39 cos(alpha3) - inner_product3/model3 == 0];
40 vars = [x y];
41 [result_x1,result_y1] = vpasolve(egns([1 2],1),vars,ini);
42 [result_x2,result_y2] = vpasolve(egns([2 3],1),vars,ini);
43 [result_x3,result_y3] = vpasolve(egns([1 3],1),vars,ini);
44 result_x = (result_x1+result_x2+result_x3)/3;
45 result_y = (result_y1+result_y2+result_y3)/3;
46 if isscalar(result_x) % 判断 $result\_x$ 是不是标量
47 error1 = 1/2*((result_x - ini(1))^2 + (result_y - ini(2))
    ^2); % 使用方差描述误差大小
48 if error > error1
49 error = error1;
50 temp_xy = [result_x,result_y];
51 end

```

```

52 end
53 end
54 output = temp_xy;
55 end

```

## 附录 D 问题 1.1 源代码 2

"shumo\_B\_1\_12\_test.m"

```

1  %% 数据导入
2  clear;clc
3  load('FY.mat')
4  [x,y] = pol2cart(FY(:,2)*pi/180,FY(:,1));
5  FY_cart = [x,y];
6
7  %% 目标位置
8  theta = 0:40:320;
9  pos_initial = [repmat(100,9,1),theta'];
10 pos_initial = [[0,0];pos_initial]; % 极坐标
11 [x,y] = pol2cart(pos_initial(:,2)*pi/180,pos_initial(:,1));
12 pos_ini_cart = [x,y]; % 直角坐标
13
14 %% 验证
15 FYsend1 = [0 0]; % FY0
16 FYsend2 = [100 0]; % FY1
17 FYsend3 = pos_ini_cart(3,:);
18 FYget = FY_cart(4:end,:); % 得到除发射点之外的其他无人机的
    坐标
19
20 % 计算出  $\alpha_{1-3}$ 
21 alpha_info = zeros(7,3);
22 for i = 1:7

```

```

23 alpha_info(i,:) = comp_alpha(FYsend1,FYsend2,FYsend3,FYget(
    i,:));
24 end
25
26 pos_ini_cart1 = pos_ini_cart(4:end,:);
27
28 % 定位
29 pos_test = zeros(7,2);
30
31 for i = 1:7
32 pos_test(i,:) = shumo_B_1_12(FYsend1,FYsend2,FYsend3,
    alpha_info(i,1),alpha_info(i,2),alpha_info(i,3),
    pos_ini_cart1(i,:));
33 end
34
35 %% 误差
36 d = sqrt(sum((pos_test - FYget).^2,2));
37 ex = abs((pos_test(:,1) - FYget(:,1))./FYget(:,1));
38 ey = abs((pos_test(:,2) - FYget(:,2))./FYget(:,2));
39 e = (ex + ey)/2;
40 E = [ex,ey,e];
41 b = bar(E);
42 ex_point_x = b(1).XEndPoints;
43 ex_point_y = b(1).YEndPoints;
44 ex_data = string(b(1).YData);
45
46 legend('横坐标相对偏差','纵坐标相对偏差','平均相对偏差','
    fontsize',15)
47 figure(2)
48 bar(d)
49 legend('距离偏差','fontsize',15)
50

```

```

51 %% 保存结果
52 T=cell(1,10);T{1}="x的理想值";T{2}="y的理想值";T{3}="x的计
    算值";T{4}="y的计算值";
53 T{5}="x的相对误差";T{6}="y的相对误差";T{7}="坐标相对误差";T
    {8}="距离";
54 writecell(T,"检验.xlsx","Range","a1:j1");
55 T=[FYget(:,1),FYget(:,2),pos_test(:,1),pos_test(:,2),ex,ey,
    e,d];
56 writematrix(T,"检验.xlsx","range","a2:j7");

```

## 附录 E 问题 1.2 源代码 1

"shumo\_B\_22.m"

```

1 clear;clc
2 %% 理想位置
3 theta = 0:40:320;
4 pos_initial = [repmat(100,9,1),theta'];
5 pos_initial = [[0,0];pos_initial]; % 极坐标
6 [x,y] = pol2cart(pos_initial(:,2)*pi/180,pos_initial(:,1));
7 pos_ini_cart = [x,y]; % 直角坐标
8
9 %% 发射信号的位置
10 FYsend1 = [0,0];
11 FYsend2 = [100,0];
12
13 %% 接收信号的无人机知道的信息：三个角度 $\alpha_1-3$ ，自己的编
    号，自己的目标位置
14 % 角度
15 alpha1 = 10;
16 alpha2 = 20;
17 alpha3 = 30;

```

```

18 alpha = [[alpha1;alpha2;alpha3],zeros(3,2)];
19 % 自己的编号
20 FYget_code = 8;
21 % 目标位置
22 ini = pos_ini_cart(8,:);
23
24 %% 通过其中一个角是 $20n$ , 来确定第三个发射点可能的两个位置
25 % 判断第三个无人机的位置
26 alpha_compare = abs(alpha(:,1) - 20);
27 [m,n] = sort(alpha_compare);
28 alpha = alpha(n,:); % 将最接近 $20$ 的角度 $n$ 的角度放在第一位
29 number = (alpha(1,1) + (20 - alpha(1,1)))/20; % 针对最接近
     $20n$ 的角求出 $n$ 
30 FYsend3_code = [1 + number + 1,10 - number + 1]; %得出第三
    个教的两种可能的编号
31
32 %% 根据角度关系确定具体是哪个编号
33 p = mod((FYget_code - 1 + 4.5),9);
34 temp_alpha_1_0 = abs(p - 1)*20; % 编号 $1-0$ 对应的夹角
35 % 1、假定是 $n1$ 
36 n1 = FYsend3_code(1);
37 temp_alpha_n1_0 = abs(p - (n1 - 1))*20; % 编号 $n1-0$ 对应的夹
    角
38 % 2、假定是 $n2$ 
39 n2 = FYsend3_code(2);
40 temp_alpha_n2_0 = abs(p - (n2 - 1))*20; % 编号 $n2-0$ 对应的夹
    角
41
42 % 求出 $n1-0$ 对应的夹角和两个已知角度的偏差
43 dist1 = abs(temp_alpha_n1_0 - alpha(2,1));
44 dist2 = abs(temp_alpha_n1_0 - alpha(3,1));
45

```

```

46 % 求出n2-0对应的夹角和两个已知角度的偏差
47 dist3 = abs(temp_alpha_n2_0 - alpha(2,1));
48 dist4 = abs(temp_alpha_n2_0 - alpha(3,1));
49
50 % 断定是哪一种情况
51 dist = [dist1;dist2;dist3;dist4];
52 [~,n] = min(dist); % 求出最小的差值
53 if n == 1
54     alpha(2,[2 3]) = [n1,1];
55     alpha(1,[2 3]) = [n1,2];
56     alpha(3,[2 3]) = [1,2];
57     FYsend3 = pos_ini_cart(n1,:);
58 elseif n == 2
59     alpha(3,[2 3]) = [n1,1];
60     alpha(1,[2 3]) = [n1,2];
61     alpha(2,[2 3]) = [1,2];
62     FYsend3 = pos_ini_cart(n1,:);
63 elseif n == 3
64     alpha(2,[2 3]) = [n2,1];
65     alpha(1,[2 3]) = [n2,2];
66     alpha(3,[2 3]) = [1, 2];
67     FYsend3 = pos_ini_cart(n2,:);
68 elseif n == 4
69     alpha(3,[2 3]) = [n2,1];
70     alpha(1,[2 3]) = [n2,2];
71     alpha(2,[2 3]) = [1,2];
72     FYsend3 = pos_ini_cart(n2,:);
73 end
74 alpha_info = alpha(:,[2 3 1]);
75 FYsend = [FYsend1;FYsend2;FYsend3];
76 %% 定位
77 syms x y

```

```

78 temp_xy1_1 = FYsend(alpha_info(1,1),:); % 记录坐标
79 temp_xy1_2 = FYsend(alpha_info(1,2),:);
80 alpha1 = alpha_info(1,3)*pi/180;
81 [temp1_up,mod1] = inner_product_model(temp_xy1_1,temp_xy1_2
    ,[x y]);
82
83 temp_xy2_1 = FYsend(alpha_info(2,1),:);
84 temp_xy2_2 = FYsend(alpha_info(2,2),:);
85 alpha2 = alpha_info(2,3)*pi/180;
86 [temp2_up,mod2] = inner_product_model(temp_xy2_1,temp_xy2_2
    ,[x y]);
87
88 temp_xy3_1 = FYsend(alpha_info(3,1),:);
89 temp_xy3_2 = FYsend(alpha_info(3,2),:);
90 alpha3 = alpha_info(3,3)*pi/180;
91 [temp3_up,mod3] = inner_product_model(temp_xy3_1,temp_xy3_2
    ,[x y]);
92
93 egns = [cos(alpha1) - temp1_up/mod1 == 0;
94 cos(alpha2) - temp2_up/mod2 == 0;
95 cos(alpha3) - temp3_up/mod3 == 0
96 ];
97 vars = [x,y];
98 [result_x1,result_y1] = vpasolve(egns([1 2],1),vars,ini);
99 [result_x2,result_y2] = vpasolve(egns([2 3],1),vars,ini);
100 [result_x3,result_y3] = vpasolve(egns([1 3],1),vars,ini);
101 result_x = (result_x1+result_x2+result_x3)/3;
102 result_y = (result_y1+result_y2+result_y3)/3;

```

## 附录 F 问题 1.2 源代码 2



```

1 %% 问题一第二问验证
2 % 针对问题一第二问中通过已知0和1是发射点，还知道自己的编号，
   以及知道三个角的大小，
3 % 进而确定三个角的边与发射点的对应关系的验证
4 clear;clc
5
6 %% 数据导入
7 load('FY.mat')
8 [x,y] = pol2cart(FY(:,2)*pi/180,FY(:,1));
9 FY_cart = [x,y];
10 FYget = FY(4:end,:);
11
12 %% 我们假定0 1 2是发射点，使用问题一第三问中各点的位置信息
   计算出应当受到的角度，进行仿真
13 % 仿真收到的角度
14 FYsend1 = [0,0];
15 FYsend2 = [100,0];
16 FYsend3 = FY_cart(3,:);
17 alpha_info = zeros(7,3);
18 for i = 1:7
19 alpha_info(i,:) = comp_alpha(FYsend1,FYsend2,FYsend3,FYget(
   i,:));
20 end
21
22 %% 理想位置
23 theta = 0:40:320;
24 pos_initial = [repmat(100,9,1),theta'];
25 pos_initial = [[0,0];pos_initial]; % 极坐标
26 [x,y] = pol2cart(pos_initial(:,2)*pi/180,pos_initial(:,1));
27 pos_ini_cart = [x,y]; % 直角坐标
28

```

```

29 %% 根据收到的角度，以及0 1 编号已知的情况下，判断第三个发射
    点是不是编号为2的无人机
30 temp_FYsend3 = zeros(7,1);
31
32 for i = 1:7
33 % 接收信号的无人机知道的信息：三个角度 $\alpha_1-3$ ，自己的编
    号，自己的目标位置
34 % 角度
35 alpha1 = alpha_info(i,1)*180/pi;
36 alpha2 = alpha_info(i,2)*180/pi;
37 alpha3 = alpha_info(i,3)*180/pi;
38 alpha = [[alpha1;alpha2;alpha3],zeros(3,2)];
39 % 自己的编号
40 FYget_code = 8;
41 % 目标位置
42 ini = pos_ini_cart(8,:);
43
44 % 通过其中一个角是 $20n$ ，来确定第三个发射点可能的两个位置
45 % 判断第三个无人机的位置
46 alpha_compare = abs(alpha(:,1) - 20);
47 [m,n] = sort(alpha_compare);
48 alpha = alpha(n,:); % 将最接近20的角度 $n$ 的角度放在第一位
49 number = (alpha(1,1) + (20 - alpha(1,1)))/20; % 针对最接近
     $20n$ 的角求出 $n$ 
50 FYsend3_code = [1 + number + 1,10 - number + 1]; %得出第三
    个教的两种可能的编号
51
52 % 根据角度关系确定具体是哪个编号
53 p = mod((FYget_code - 1 + 4.5),9);
54 temp_alpha_1_0 = abs(p - 1)*20; % 编号1-0对应的夹角
55 % 1、假定是 $n1$ 
56 n1 = FYsend3_code(1);

```

```

57 temp_alpha_n1_0 = abs(p - (n1 - 1))*20; % 编号n1-0对应的夹
    角
58 % 2、假定是n2
59 n2 = FYsend3_code(2);
60 temp_alpha_n2_0 = abs(p - (n2 - 1))*20; % 编号n2-0对应的夹
    角
61
62 % 求出n1-0对应的夹角和两个已知角度的偏差
63 dist1 = abs(temp_alpha_n1_0 - alpha(2,1));
64 dist2 = abs(temp_alpha_n1_0 - alpha(3,1));
65
66 % 求出n2-0对应的夹角和两个已知角度的偏差
67 dist3 = abs(temp_alpha_n2_0 - alpha(2,1));
68 dist4 = abs(temp_alpha_n2_0 - alpha(3,1));
69
70 % 断定是哪一种情况
71 dist = [dist1;dist2;dist3;dist4];
72 [~,n] = min(dist); % 求出最小的差值
73 if n == 1
74     temp_FYsend3(i) = n1;
75 elseif n == 2
76     temp_FYsend3(i) = n1;
77 elseif n == 3
78     temp_FYsend3(i) = n2;
79 elseif n == 4
80     temp_FYsend3(i) = n2;
81 end
82 end

```

## 附录 G 问题 1.3 源代码

"shumo\_B\_1\_3.m"

```
1 clear;clc
2 %% 导入实际位置
3 load('FY.mat') % FY中的位置信息是极坐标表示
4 [x,y] = pol2cart(FY(:,2)*pi/180,FY(:,1));
5 FY_cart = [x,y]; % 转化为直角坐标
6 %% 目标位置
7 theta = 0:40:320;
8 pos_initial = [repmat(100,9,1),theta'];
9 pos_initial = [[0,0];pos_initial]; % 极坐标
10 [x,y] = pol2cart(pos_initial(:,2)*pi/180,pos_initial(:,1));
11 pos_ini_cart = [x,y]; % 直角坐标
12 %% 保存每次更新后的位置
13 pos_renew = FY_cart;
14 count = 0; % 记录迭代的次数
15 J_val = sum((FY_cart - pos_ini_cart).^2,2); % 当J_val小于
16 while max(J_val) > 0.01
17     count = count + 1;
18     % 找出3个发送位置的坐标
19     error = sum((FY_cart - pos_ini_cart).^2,2); % 计算出实际位
        置偏离目标位置的距离
20     [m,p] = sort(error);
21     FYsend1 = FY_cart(p(1),:); % 选取偏差最小的三个位置作为发射
        位置
22     FYsend2 = FY_cart(p(2),:);
23     FYsend3 = FY_cart(p(3),:);
24     FYsend = [FYsend1;FYsend2;FYsend3]; % 实际发射位置的坐标
25
26     FYsend4 = pos_ini_cart(p(1:3),:); % 理想发射位置
27
28 % 由于条件限制，无法测量接收信号的无人机接收到的方向信息，
    所以采用根据实际位置计算的方式进行仿真
```

```

29 FYget = FY_cart(p(4:end),:); % 找到接收信号的无人机的实际坐
    标, 这些无人机的编号是n(4:end)
30 alpha_info = [p(4:end),zeros(7,3)]; % 四列, 分别是无人机编
    号, 以及alpha1-3
31 for i = 1:7
32     alpha_info(i,2:end) = comp_alpha(FYsend1,FYsend2,FYsend3,
        FYget(i,:));
33 end
34
35 % 根据方向信息确定接收信号的无人机的位置
36 pos_comp = [p(4:end),zeros(7,2)]; % 第一列是无人机编号, 之
    后是对应的x和y坐标
37 FYsend_perm = perms([1 2 3]); % 给出1 2 3这三个发射位置编号
    的排列
38 n = size(FYsend_perm,1);
39 for j = 1:7 % 每次考察的是编号为p(j + 3)的无人机
40     error = +inf; % 设定所求出的位置和目标位置误差的初始值为正
        无穷
41     temp_xy = [];
42     syms x y
43     alpha = alpha_info(j,2:end);
44     ini = pos_ini_cart(p(j + 3),:); % 目标位置
45     for i = 1:n
46         temp_xy_code1 = FYsend_perm(i,[1 2]); % alpha1对应的两个发
            射点编号
47         temp_xy_1_1 = FYsend4(temp_xy_code1(1),:);
48         temp_xy_1_2 = FYsend4(temp_xy_code1(2),:);
49         [inner_product1,model1] = inner_product_model(temp_xy_1_1,
            temp_xy_1_2,[x y]);
50
51     temp_xy_code2 = FYsend_perm(i,[2 3]); % alpha2对应的两个发
        射点编号

```

```

52 temp_xy_2_1 = FYsend4(temp_xy_code2(1),:);
53 temp_xy_2_2 = FYsend4(temp_xy_code2(2),:);
54 [inner_product2,model2] = inner_product_model(temp_xy_2_1,
    temp_xy_2_2,[x y]);
55
56 temp_xy_code3 = FYsend_perm(i,[1 3]); % alpha3对应的两个发
    射点编号
57 temp_xy_3_1 = FYsend4(temp_xy_code3(1),:);
58 temp_xy_3_2 = FYsend4(temp_xy_code3(2),:);
59 [inner_product3,model3] = inner_product_model(temp_xy_3_1,
    temp_xy_3_2,[x y]);
60 % 定位
61 egns = [cos(alpha(1)) - inner_product1/model1 == 0;
62 cos(alpha(2)) - inner_product2/model2 == 0;
63 cos(alpha(3)) - inner_product3/model3 == 0;];
64 vars = [x y];
65 [result_x1,result_y1] = vpasolve(egns([1 2],1),vars,ini);
66 [result_x2,result_y2] = vpasolve(egns([2 3],1),vars,ini);
67 [result_x3,result_y3] = vpasolve(egns([1 3],1),vars,ini);
68 result_x = (result_x1+result_x2+result_x3)/3;
69 result_y = (result_y1+result_y2+result_y3)/3;
70 if isscalar(result_x) % 判断 $result\_x$ 是不是标量
71 error1 = 1/2*((result_x - ini(1))^2 + (result_y - ini(2))
    ^2); % 使用方差描述误差大小
72 if error > error1
73 error = error1;
74 temp_xy = [result_x,result_y];
75 end
76 end
77 end
78 pos_comp(j,2:end) = temp_xy;
79 end

```

```

80 u = pos_comp(:,1); % 对u中对应的行进行调整
81 move = pos_comp(:,2:3) - pos_ini_cart(u,:); % 需要移动的距离
82 temp = FY_cart(u,:) - move; % 移动之后的距离
83 FY_cart(u,:) = temp; % 对实际位置进行更新
84 pos_renew(:, :, count + 1) = FY_cart; % 记录每次调整位置之后的坐标
85 J_val = sum((FY_cart - pos_ini_cart).^2,2); % 对J_val进行更新
86 end
87
88 %% 作图
89 for i = 1:6
90 figure(i)
91 hold on;
92 %作出圆心坐标为(0,0)、半径为100的圆
93 viscircles([0 0],100,'Color','g','linewidth',1,'LineStyle','--');
94 axis([-115,115,-115,115]);
95 axis equal
96
97 plot(pos_renew(:,1,i),pos_renew(:,2,i),'.','MarkerSize',20)
98 ;
99 end

```

## 附录 H 问题 2 源代码

"shumo\_B\_222.m"

```

1 %% 数据导入，由于题目中没有给出数据，这里认为设定目标位置
2 clear;clc
3

```

```

4  %% 模拟目标位置
5  % 第一行
6  h = 0;
7  FY2_1 = [[0,h];[200,h];[400,h];[600,h];[800,h]];
8  % 第二行
9  h = 100*sqrt(3);
10 FY2_2 = [[100,h];[300,h];[500,h];[700,h]];
11 % 第三行
12 h = 200*sqrt(3);
13 FY2_3 = [[200,h];[400,h];[600,h]];
14 % 第四行
15 h = 300*sqrt(3);
16 FY2_4 = [[300,h];[500,h]];
17 % 第五行
18 h = 400*sqrt(3);
19 FY2_5 = [400,h];
20
21 FY2_ini = [FY2_1;FY2_2;FY2_3;FY2_4;FY2_5]; % 直角坐标
22
23 %% 模拟实际位置,给这15个位置的初始位置加一个扰动项
24 FY_cart = FY2_ini + [(randi(10,15,1) - 5),(randi(10,15,1) -
    5)];
25 FY_cart([11,12],:) = FY2_ini([11,12],:); % 保证编号位11和12
    的位置的无人机的坐标是正确的
26
27 %% 求解1,先迭代调整由7 8 10 11 12 13 14这几个点确定的圆
28
29 % 将上述圆对应的数据提取
30 temp_p = [11 12 14 13 10 7 8]; % 记录这个圆中各点的编号,除
    了11和12
31 % 圆上各点的实际位置
32 circle_actual = FY_cart(temp_p,:);

```



```

33
34 % 圆上各点的理想位置，也是目标位置
35 circla_target = FY2_ini(temp_p,:);
36
37 % 前两个发射点的实际位置
38 FYsend1_actual = circle_actual(1,:);
39 FYsend2_actual = circle_actual(2,:);
40
41 % 前两个发射点的目标位置
42 FYsend1_target = circla_target(1,:);
43 FYsend2_target = circla_target(2,:);
44
45 % 对当前的圆进行迭代
46 J_val = sum((circle_actual(3:end,:) - circla_target(3:end
    ,:)).^2,2); % 各点偏离距离的平方
47
48 iter = 1;
49 pos_iter_actual = circle_actual; % 用于记录每次迭代后的位置
    坐标
50
51 while max(J_val) >6
52     iter = iter + 1;
53
54 % 找到偏离实际位置最小距离的三个点,有两个是固定的, 11和12,
    再取一个最小的
55 [m,n] = sort(J_val);
56
57 % 找到第三个发射点的实际位置
58 FYsend3_actual = circle_actual(n(1) + 2,:);
59
60 % 找到第三个发射点的理想位置
61 FYsend3_target = circla_target(n(1) + 2,:);

```

```

62 FYsend = [FYsend1_target;FYsend2_target;FYsend3_target];
63
64 % 找到其余4个等待接收信息的无人机的实际位置
65 FYget_4_actual = circle_actual(n(2:end) + 2,:);
66
67 % 找到其余4个等待接收信息的无人机的目标位置
68 FYget_4_target = circla_target(n(2:end) + 2,:);
69
70 % 利用三个发射点的实际位置,以及四架接收信息的无人机的实际位置进行方位信息的仿真,单位是弧度
71 alpha_info = zeros(4,3);
72 for i = 1:4
73     alpha_info(i,:) = comp_alpha(FYsend1_actual,FYsend2_actual,
74                                   FYsend3_actual,FYget_4_actual(i,:));
75 end
76
77 % 每个点有了方向信息以及发送位置的理想坐标,计算其余四个点的位置
78 FYsend_perm = perms([1 2 3]);
79 r = size(FYsend_perm,1);
80 pos_new = zeros(4,2);% 用于保留每次迭代时更新的位置
81 for j = 1:4 % 以此计算接收信号的四个无人机的位置
82     error = +inf; % 设定所求出的位置和目标位置误差的初始值为正无穷
83     temp_xy = []; % 记录FYsend_perm中误差最小的情况对应位置
84     alpha = alpha_info(j,:); % 得出当前无人机对应的三个角度
85     ini = FYget_4_target(j,:); % 得出当前无人机对应的目标位置
86     syms x y
87     for i = 1:r % 对每一个无人机对应的六种情况进行讨论
88         disp(i)
89

```

```

90 % 进行角度的选择，原则是选择 $\alpha_1-3$ 中最大的两个，即择取序号
    为 $v(2)$ 和 $v(3)$ 的两个角
91 [u,v] = sort(alpha);
92
93 if v(1) == 1 && v(3) == 2 || v(1) == 2 && v(3) == 1
94
95 %  $\alpha_1$ 对应的两个发射点是 $temp\_xy\_1\_1$ 和 $temp\_xy\_1\_2$ 
96 temp_xy_code1 = FYsend_perm(i,[1 2]); %  $\alpha_1$ 对应的两个发
    射点编号
97 temp_xy_1_1 = FYsend(temp_xy_code1(1),:);
98 temp_xy_1_2 = FYsend(temp_xy_code1(2),:);
99 [inner_product1,model1] = inner_product_model(temp_xy_1_1,
    temp_xy_1_2,[x y]);
100
101 %  $\alpha_2$ 对应的两个发射点是 $temp\_xy\_2\_1$ 和 $temp\_xy\_2\_2$ 
102 temp_xy_code2 = FYsend_perm(i,[2 3]); %  $\alpha_2$ 对应的两个发
    射点编号
103 temp_xy_2_1 = FYsend(temp_xy_code2(1),:);
104 temp_xy_2_2 = FYsend(temp_xy_code2(2),:);
105 [inner_product2,model2] = inner_product_model(temp_xy_2_1,
    temp_xy_2_2,[x y]);
106
107
108 egns = [cos(alpha(1)) - inner_product1/model1 == 0;
109 cos(alpha(2)) - inner_product2/model2 == 0];
110 elseif v(1) == 1 && v(3) == 3 || v(1) == 3 && v(3) == 1
111
112 %  $\alpha_1$ 对应的两个发射点是 $temp\_xy\_1\_1$ 和 $temp\_xy\_1\_2$ 
113 temp_xy_code1 = FYsend_perm(i,[1 2]); %  $\alpha_1$ 对应的两个发
    射点编号
114 temp_xy_1_1 = FYsend(temp_xy_code1(1),:);
115 temp_xy_1_2 = FYsend(temp_xy_code1(2),:);

```

```

116 [inner_product1,model1] = inner_product_model(temp_xy_1_1,
        temp_xy_1_2,[x y]);
117
118 % alpha3对应的两个发射点是 temp_xy_3_1和 temp_xy_3_2
119 temp_xy_code3 = FYsend_perm(i,[1 3]); % alpha3对应的两个发
        射点编号
120 temp_xy_3_1 = FYsend(temp_xy_code3(1),:);
121 temp_xy_3_2 = FYsend(temp_xy_code3(2),:);
122 [inner_product3,model3] = inner_product_model(temp_xy_3_1,
        temp_xy_3_2,[x y]);
123
124 egns = [cos(alpha(1)) - inner_product1/model1 == 0;
125 cos(alpha(3)) - inner_product3/model3 == 0];
126 elseif v(1) == 2 && v(3) == 3 || v(1) == 3 && v(3) == 2
127
128 % alpha2对应的两个发射点是 temp_xy_2_1和 temp_xy_2_2
129 temp_xy_code2 = FYsend_perm(i,[2 3]); % alpha2对应的两个发
        射点编号
130 temp_xy_2_1 = FYsend(temp_xy_code2(1),:);
131 temp_xy_2_2 = FYsend(temp_xy_code2(2),:);
132 [inner_product2,model2] = inner_product_model(temp_xy_2_1,
        temp_xy_2_2,[x y]);
133
134 % alpha3对应的两个发射点是 temp_xy_3_1和 temp_xy_3_2
135 temp_xy_code3 = FYsend_perm(i,[1 3]); % alpha3对应的两个发
        射点编号
136 temp_xy_3_1 = FYsend(temp_xy_code3(1),:);
137 temp_xy_3_2 = FYsend(temp_xy_code3(2),:);
138 [inner_product3,model3] = inner_product_model(temp_xy_3_1,
        temp_xy_3_2,[x y]);
139
140 egns = [cos(alpha(2)) - inner_product2/model2 == 0;

```

```

141 cos(alpha(3)) - inner_product3/model3 == 0];
142 end
143 vars = [x ,y];
144 [result_x,result_y] = vpasolve(egns,vars,ini);
145 if isscalar(result_x) % 判断result_x是不是标量
146 error1 = 1/2*((result_x - ini(1))^2 + (result_y - ini(2))
    ^2); % 使用方差描述误差大小
147 if error >= error1
148 error = error1;
149 temp_xy = [result_x,result_y];
150 end
151 end
152 end
153 pos_new(j,:) = temp_xy; % 计算出来的位置
154 end
155
156 % 更新后的坐标: 原有位置 + (理想位置 - pos_new)
157 temp11 = FYget_4_actual + (FYget_4_target - pos_new);
158 circle_actual(n(2:end) + 2,:) = temp11;
159 pos_iter_actual(:, :, iter) = circle_actual;
160 J_val = sum((circle_actual(3:end,:) - circla_target(3:end
    ,:)).^2,2); % 各点偏离距离的平方
161 end
162
163
164 %% 将求解一得出的结果保存在pos_iter_actual.mat中
165 load('pos_iter_actual.mat')
166 temp = pos_iter_actual(:, :, 3);
167 %% 求解2, 确定编号为3 4 8 9的无人机的坐标
168
169 %% 可以确定
170 FYsend1_actual = temp(3,:);

```

```

171 FYsend2_actual = temp(1,:);
172 FYsend3_actual = temp(7,:);
173
174 % 实际坐标
175 pos_actual = FY_cart([3 4 8 9],:);
176 % 目标位置
177 pos_target = FY2_ini([3 4 8 9],:);
178
179 %% 根据发送位置的实际坐标来求三个发射点到3 4 8 9这几架无人
    机的坐标
180
181 % 通过实际位置得出角度
182 alpha_info = zeros(4,3);
183 for i = 1:4
184     alpha_info(i,:) = comp_alpha(FYsend1_actual,FYsend2_actual,
        FYsend3_actual,pos_actual(i,:));
185 end
186
187 % 对无人机7 8 11 12 以上面得出的发送点位置以及收到的方位信
    息
188 pos_comp2 = zeros(4,2);
189 for i = 1:4
190     pos_comp2(i,:) = shumo_B_1_12(FYsend1_actual,FYsend2_actual
        ,FYsend3_actual,alpha_info(i,1),alpha_info(i,2),
        alpha_info(i,3),pos_target(i,:));
191 end
192
193 %% 求解3, 确定6 2 3 的位置
194
195 %% 可以确定
196 FYsend1_actual = temp(6,:);
197 FYsend2_actual = temp(1,:);

```

```

198 FYsend3_actual = temp(2,:);
199
200 % 实际坐标
201 pos_actual = FY_cart([6 2 3],:);
202 % 目标位置
203 pos_target = FY2_ini([6 2 3],:);
204
205 %% 根据发送位置的实际坐标来求三个发射点到3 4 8 9这几架无人
    机的坐标
206
207 % 通过实际位置得出角度
208 alpha_info = zeros(3,3);
209 for i = 1:3
210     alpha_info(i,:) = comp_alpha(FYsend1_actual,FYsend2_actual,
        FYsend3_actual,pos_actual(i,:));
211 end
212
213 % 对无人机7 8 11 12 以上面得出的发送点位置以及收到的方位信
    息
214 pos_comp3 = zeros(3,2);
215 for i = 1:3
216     pos_comp3(i,:) = shum_B_1_12(FYsend1_actual,FYsend2_actual
        ,FYsend3_actual,alpha_info(i,1),alpha_info(i,2),
        alpha_info(i,3),pos_target(i,:));
217 end
218
219 %% 求解4, 求编号为5的无人机
220 %% 可以确定
221 FYsend1_actual = FY_cart(4,:);
222 FYsend2_actual = temp(2,:);
223 FYsend3_actual = temp(3,:);
224

```

```

225 % 实际坐标
226 pos_actual = FY_cart(5,:);
227 % 目标位置
228 pos_target = FY2_ini(5,:);
229
230 % 通过实际位置得出角度
231 alpha_info = comp_alpha(FYsend1_actual,FYsend2_actual,
    FYsend3_actual,pos_actual);
232
233 % 对无人机5 以上面得出的发送点位置以及收到的方位信息
234 pos_comp4 = shumob_1_12(FYsend1_actual,FYsend2_actual,
    FYsend3_actual,alpha_info(1),alpha_info(2),alpha_info(3),
    pos_target);
235
236 %% 求解5, 确定编号为15无人机的坐标
237 %% 可以确定,
238 FYsend1_actual = temp(3,:);
239 FYsend2_actual = temp(1,:);
240 FYsend3_actual = temp(5,:);
241
242 % 实际坐标
243 pos_actual = FY_cart(15,:);
244 % 目标位置
245 pos_target = FY2_ini(15,:);
246
247 % 通过实际位置得出角度
248 alpha_info = comp_alpha(FYsend1_actual,FYsend2_actual,
    FYsend3_actual,pos_actual);
249
250 % 对无人机5 以上面得出的发送点位置以及收到的方位信息
251 pos_comp5 = shumob_1_12(FYsend1_actual,FYsend2_actual,
    FYsend3_actual,alpha_info(1),alpha_info(2),alpha_info(3),

```



```

pos_target);
252
253
254 %%
255 load('pos_iter_actual.mat') % 11 12 14 13 10 7 8
256 load('pos_comp2.mat') % 3 4 8 9
257 load('pos_comp3.mat') % 6 2 3
258 load('pos_comp4.mat') % 5
259 load('pos_comp5.mat') % 15
260 load('pos_comp6.mat') % 1
261 temp = [pos_comp6;pos_comp3([2,3],:);pos_comp2(2,:);
        pos_comp4;pos_comp3(1,:);pos_iter_actual([6 7],:,3);
        pos_comp2(4,:);
262 pos_iter_actual(5,:,3);pos_iter_actual([1 2],:,3);
        pos_iter_actual([4 3],:,3);pos_comp5];
263 load('FY_cart.mat')
264 e = sqrt(sum((double(temp - FY_cart)).^2,2))./200;
265 bar(e)
266 xlabel('无人机编号')
267 ylabel('相对误差')

```