
摘要

本文针对无人机集群遂行编队飞行时调整自身定位的问题，建立几何模型，利用 cad、matlab 等工具进行演算求解，制定了在**纯方位无源定位**的方法下，无人机根据自身信息确定相对坐标以调整定位，并尽量少向外发射电磁波信号的方案。

针对问题一，根据问题描述，使用 FY00 为极点、FY01 的方向为极轴，在其高度平面建立极坐标系，每架无人机的预期位置已知，将问题转化为不同情况下根据纯方位定位确定自身坐标的问题，并使用 **AOA 三角定位算法** 进行分析。

对于第一小问，无人机需要根据接受的信号确定自身坐标。通过对第三个无人机的位置进行分类，使用正弦定理求解定位无人机的极坐标。

对于第二小问，求解至少需要多少额外信号源才能定位一个无人机，本问为 AOA 三角定位算法的成立基础，由圆周角相等性质推导得出证明。

对于第三小问，在前两个解答的基础上，选定两个无人机以及一个偏差较小的无人机构成定位基础，对其余无人机进行调整，得到局部最优解，使用引入**粒子群算法**，进行迭代，通过多次选择和调整得到最终定位。

针对问题二，无人机仍然使用纯方位无源定位的方法，首先建立与问题一相似的三角定位模型，选定两个无人机确定第三个无人机，以此确定一个平面。引入**贪心算法**，选择最近的无人机进行精确定位和调整以得到局部最优解，并进行迭代直到所有无人机都定位完成。

本模型的特色在于综合使用了纯几何证明、几何计算、粒子群算法、贪心算法等数学方法，使用 cad 绘图、利用 matlab 进行仿真求解，最终模型能够根据输入数据得到较准确输出结果，思路清晰，操作简单，有一定推广价值。

关键词： 纯方位无源定位 AOA 三角定位算法 粒子群算法 贪心算法

1.问题重述

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。为保持编队队形，拟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，同时尽可能少向外发射电磁波信号。

每架无人机所能获得的信息仅限于以自身为顶点、以其它发射源为边的角度值，确定调整方案时，需要通过这些角度值确定自身在编队中的相对位置。

问题一中，编队由 10 架无人机组成，构成圆形编队，其中 9 架无人机均匀分布在某一圆周上，另外一架无人机位于圆心。所有无人机均保持在同一高度飞行。

- (1) 位于圆心的无人机（FY00）和编队中另 2 架无人机发射信号，其余位置略有偏差的无人机被动接收信号，并且发射信号的无人机位置无偏差且编号已知。即一共有三个定位无人机，而他们不在同一条直线上。
- (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，求解需要多少额外无人机发射信号才能确定偏差无人机的坐标。
- (3) 1 架无人机位于圆心，另 9 架无人机均匀分布在半径为 100 m 的圆周上。需要根据给定的位置，设计合理的无人机位置调整方案，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。

问题二中，无人机集群并非圆形编队，而是锥形编队，直线上相邻两架无人机的间距相等，如 50 m。仍考虑纯方位无源定位的情形，设计无人机位置调整方案。

两个问题都需要两个步骤，即“定位”和“调整”，关键在于定位条件、如何定位以及调整次数。

2.模型假设

假设 1：无人机大致阵型已知，角度偏离不超过 40° ，且处于同一平面内

假设 2：每架无人机标号固定，且明确知道信源编号。

假设 3：无人机飞行不会改变其偏差位置。

假设 4：无人机偏差位置调整为瞬间发生，无调整时间干扰。

假设 5：信号传播过程为瞬时发生，无信号衰减、叠加、干涉

假设 6：无人机发射信号发生重叠时，不会对无人机的位置信息产生影响，

假设 7：无人机可精准接受信源所发信号

假设 8：无人机被动测距只能精确测得与两个发射源的夹角，但是不能知道发射机位于接收机的绝对方位

假设 9：无人机调整到精确位置后可作为新信源辅助定位

3.符号说明

$\alpha_1、\alpha_2、\alpha_3$	接受信号无人机所接受到的角度信息， 且规定 $\alpha_1 \leq \alpha_2 < \alpha_3$
$\alpha = \frac{2}{9}\pi$	FY00 无人机与圆周上两相邻无偏差无人机之间的夹角。
θ	FY00 为极点，FY01 为极轴下极坐标的角度， 且规定 $0 < \theta < \pi$
x	FY00 为极点，FY01 为极轴下极坐标的的长度
ρ	接收信号无人机的代称
$\angle abc$	以 FY0a、FY0b、FY0c 三个顶点形成的角

4.问题一模型与求解

4.1.第一问

4.1.1 模型建立及求解

设圆周半径为 R ，由无人机均匀分布可知，相邻无人机与圆心夹角为 $\alpha = \frac{2}{9}\pi$ ，以 FY00 为端点且经过 FY01 的射线为极轴建立极坐标系。不失一般性，假设在圆周上的其中一架发射信号无人机为 FY01，转化为极坐标后，其对应的极坐标为 $(R, 0)$ 。由于发射信号的无人机都是位置准确的，可推知另一架发射信号无人机的位置根据编号固定，设其编号为 FY0K，理想状态下，所有无人机均匀分布在圆周上，则其对应的坐标为 $(R, (K-1)\alpha)$ ，其中 $K \neq 0, 1$

假设接受信号的无人机 ρ 的极坐标位置为 (x, θ) ，由于剩下无人机的位置略有偏差，因此这两个参数都需要进行确定。设该无人机 ρ 与三架发射信号的无人机的夹角。并将三个角度信息进行排序，以进一步简化问题。故假设 $\alpha_1 \leq \alpha_2 < \alpha_3$ ，接下来需要通过已知信息来确定出相应接收信号无人机的极坐标 (x, θ) 。

我们将略有偏差的无人机与确定的无人机位置相连，利用正弦定理，通过联立方程组来

解得极坐标 (x, θ) 。受角度约束条件的影响，还需要对 K 的不同值进行分类讨论：

(a) 当 $K=2$ 时，两个外围确定的无人机之间没有其它无人机，其可视化分布如图 1 所示

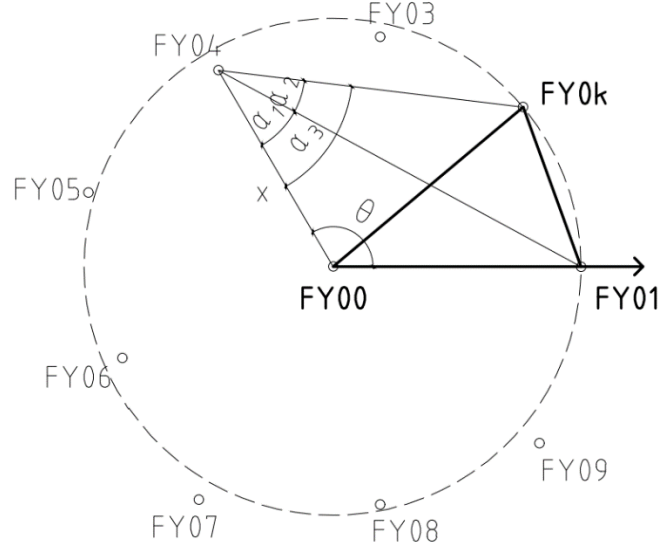


图 1-外围确定的无人机之间没有其它无人机的可视化分布

考察 FY00、FY01、FY0K 构成的三角形与 FY00、FY02、FY0K 构成的三角形，由正弦定理可得：

$$\begin{cases} \frac{\sin \alpha_1}{R} = \frac{\sin(\pi - \theta - \alpha_1)}{x} \\ \frac{\sin \alpha_3}{R} = \frac{\sin[\pi - (\theta - (K-1)\alpha + \alpha_2)]}{x} \end{cases}$$

解得：

$$\begin{cases} \theta = \tan^{-1} \frac{\sin \alpha_1 \sin \alpha_2 - \sin \alpha_1 \sin \beta_1}{\sin \alpha_1 \cos \beta_1 - \sin \alpha_2 \cos \beta_1} \\ x = \frac{\sin(\pi - \theta - \alpha_1)R}{\sin \alpha_1} \end{cases}$$

其中 $\beta_1 = \alpha_2 - (K - 1)\alpha$

(b) 当 $K=3,4,5$ 时, 需要根据 α_3 的大小, 判断有偏差的无人机是否处于两个确定位置的无人机中间; 分布情况可视化分析如图 2 所示:

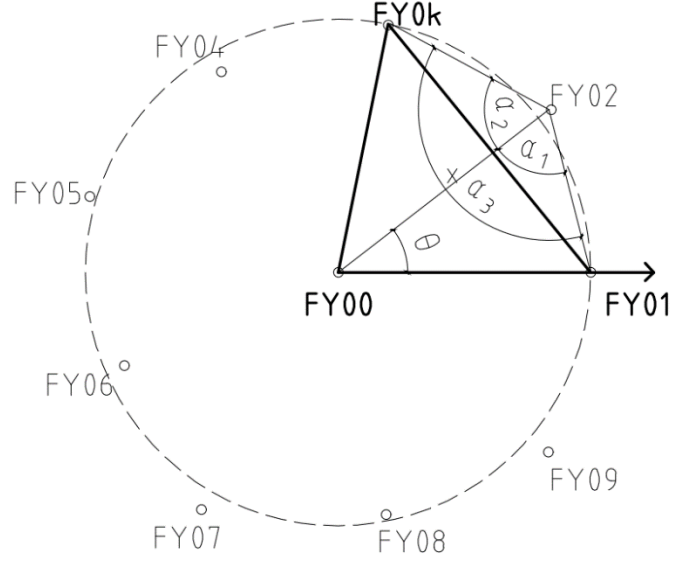


图 2 (a)有偏差无人机处于确定位置无人机中间的情况

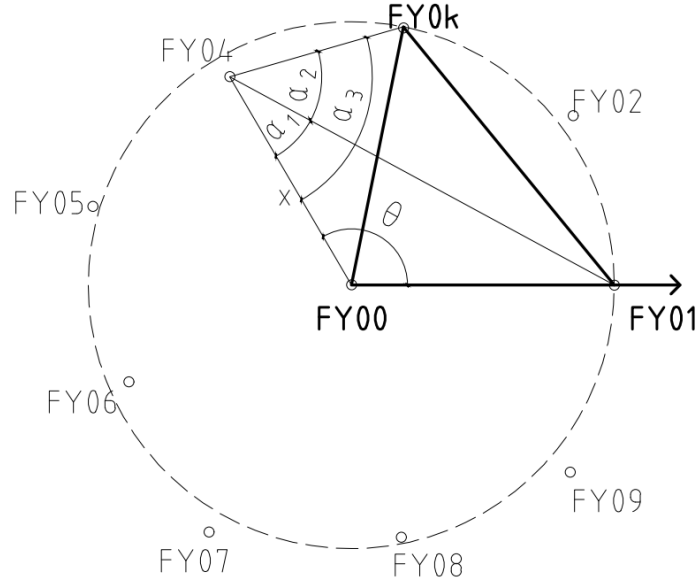


图 2 (b) 有偏差无人机处于确定位置无人机中间的情况

当两信源中间没有无人机时, 模型同 $K=2$ 时, 此处无需再过讨论, 下面重点求解两信

源中间有无人机时的情况，其可视化分布为图 3

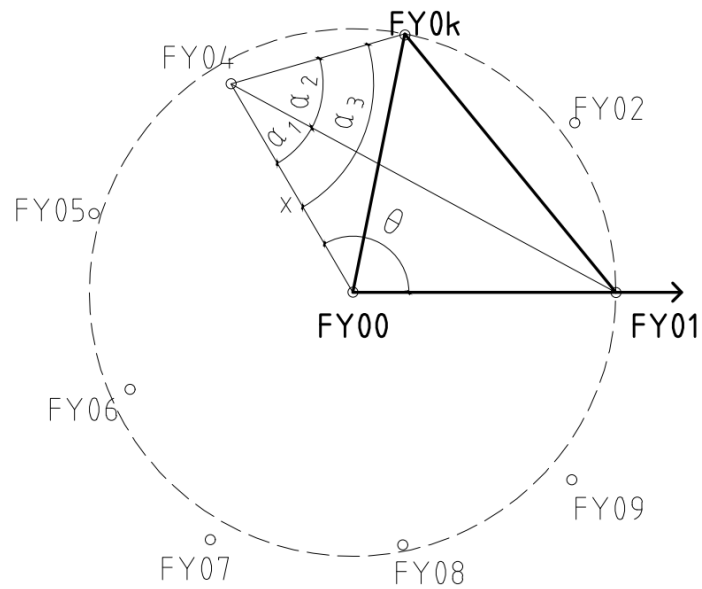


图 3

我们考察 FY00、FY01、FY0K 构成的三角形与 FY00、FY02、FY0K 构成的三角形，根据正弦定理可得：

$$\begin{cases} \frac{\sin \alpha_1}{R} = \frac{\sin(\pi - \theta - \alpha_1)}{x} \\ \frac{\sin \alpha_2}{R} = \frac{\sin[\pi - (\alpha_2 + (K-1)\alpha + \theta)]}{x} \end{cases}$$

解得：

$$\begin{cases} \theta = \tan^{-1} \frac{\sin \beta_2 \sin \alpha_1 - \sin \alpha_1 \sin \alpha_2}{\sin \alpha_1 \cos \beta_2 + \sin \alpha_2 \cos \alpha_1} \\ x = \frac{\sin(\pi - \theta - \alpha_1)R}{\sin \alpha_1} \end{cases}$$

其中 $\beta_2 = \alpha_2 + (K-1)\alpha$

(c) 当 $5 < K \leq 9$ 时, 情况与 $11-K$ ($1 < K \leq 5$) 时相对应, 只需将 $1 < K \leq 5$ 时方程中的 α 用 β 替换即可.

4.1.2 偏差范围约束

接着, 我们给出上述讨论成立的条件以及偏差范围。补充说明每个无人机偏移的最大范围, 超出该范围则使用上述算法时需要分辨无人机原本的位置。

如图，FY03 偏移范围在预期位置前后各 20°的扇形之内，并且向外偏移的位置不能超过圆周上两个发出信号无人机的连线，否则不能判断此无人机为 FY03 或是 FY04。例如图 4 所示。

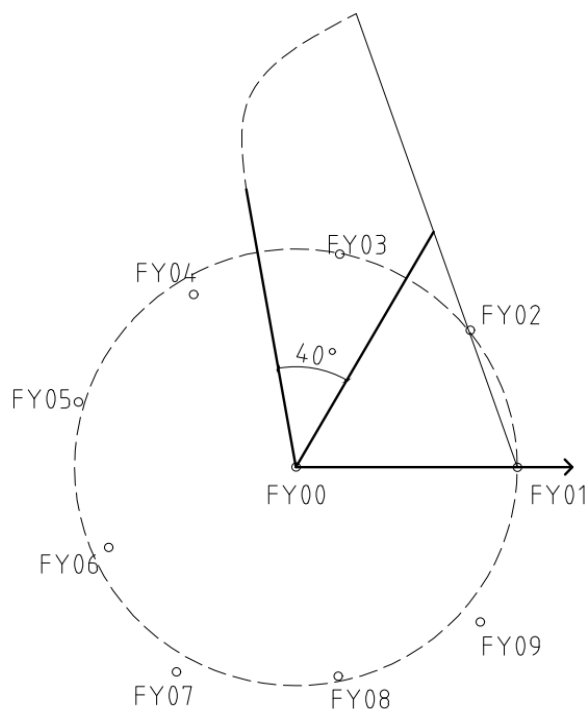


图 4

同理，在两无偏差信号发射无人机之间时，以 FY04 为例，若其偏差角度超过 20° ，则无法判断 FY05、FY04、FY03 的位置，情况更加复杂。如图 5 所示。

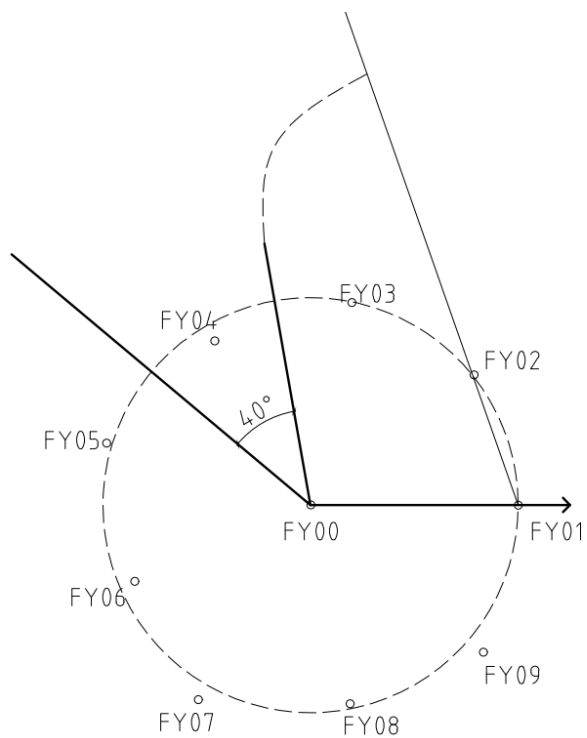


图 5

4.2.第二问

4.2.1 最少无人机个数

首先考虑仅由 FY00 和 FY01 两架无人机能否确定任意无人的位置，答案是否定的，如图 6 所示，对于 FY04 来说，由角 α_1 仅能确定其在圆 C_1 上，而不能确定其坐标。

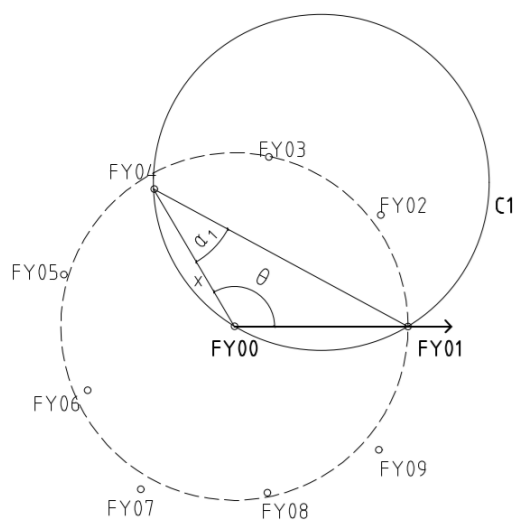


图 6

进而考虑使用额外一台无人机，如图 7 所示，加入 FY0K 后，由角 α_2 可以确定 FY04 也在圆 C_2 上，由于预期位置中 10 架无人机均不在同一直线上，故两圆必相交，且一交点为 FY00，另一交点即为 FY04；进一步考虑需要额外两个无人机的情况，由四个点确定的三个圆将带来冗余信息，故不符合。

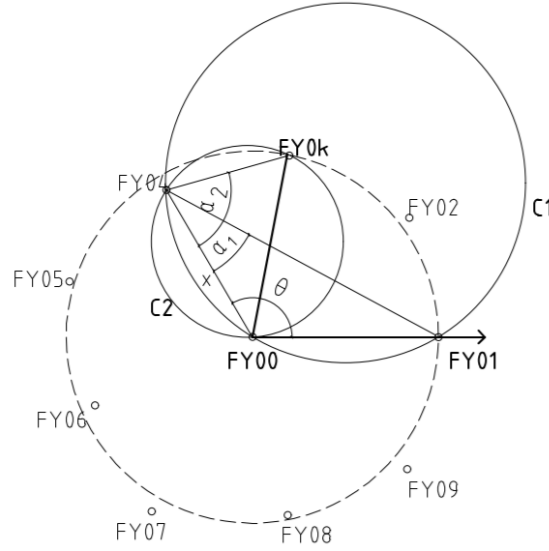


图 7

经过上述证明过程实际为 AOA 三角定位算法的基础，经过分析可知：除 FY00 和 FY01 外，还需要 1 架无人机发射信号，才能实现无人机的有效定位，这也解释 4.1 得出正确结果的原因。

4.2.2 无人机定位

因为 9 架无人机均匀分布，因此可以初始化各个编号无人机的 α 角，即角 ABO 的大小，运动过程中比较初始角与信号角的大小即可判断位置是否偏移。由 4.2.1 可知，三架编号已知的无偏差无人机已经可以定位，且 FY00、FY01 位置无偏差，因此只需要确定另外一架发射无人机的编号即可。下求另一无偏差信源的编号。

假设在多个未确定无人机 FY02 的信号，根据角度信息可能会有 FY02 和 FY09 两种情况，如下图 8 所示，当在 FY05 位置时， $\angle FY02-FY05-FY00$ 必大于 $\angle FY02-FY05-FY01$ ，在 FY06 位置时， $\angle FY09-FY06-FY00$ 必小于 $\angle FY09-FY06-FY01$ ，因此根据信号的角度信息可以确定信号接收无人机与另一编号未知信号发射无人机是同侧或异侧。根据信号角度与初始时的角度对比可推知信号发射无人机的编号，进而对比 $\angle FY02-FY05-FY00$ 、 $\angle FY02-FY05-FY01$ 、 $\angle FY05-FY02-FY01$ 三个角的当前与初始角度，分析前后偏移与左右偏移从而推算 B 该朝何方位飞行。

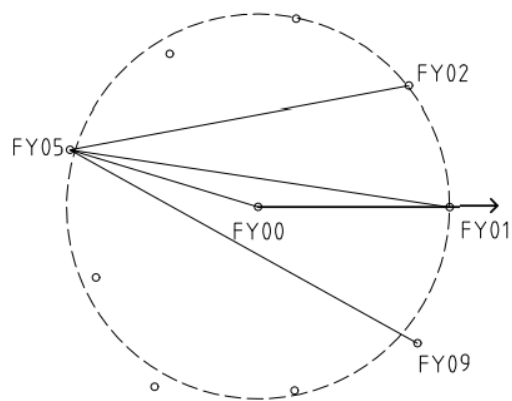
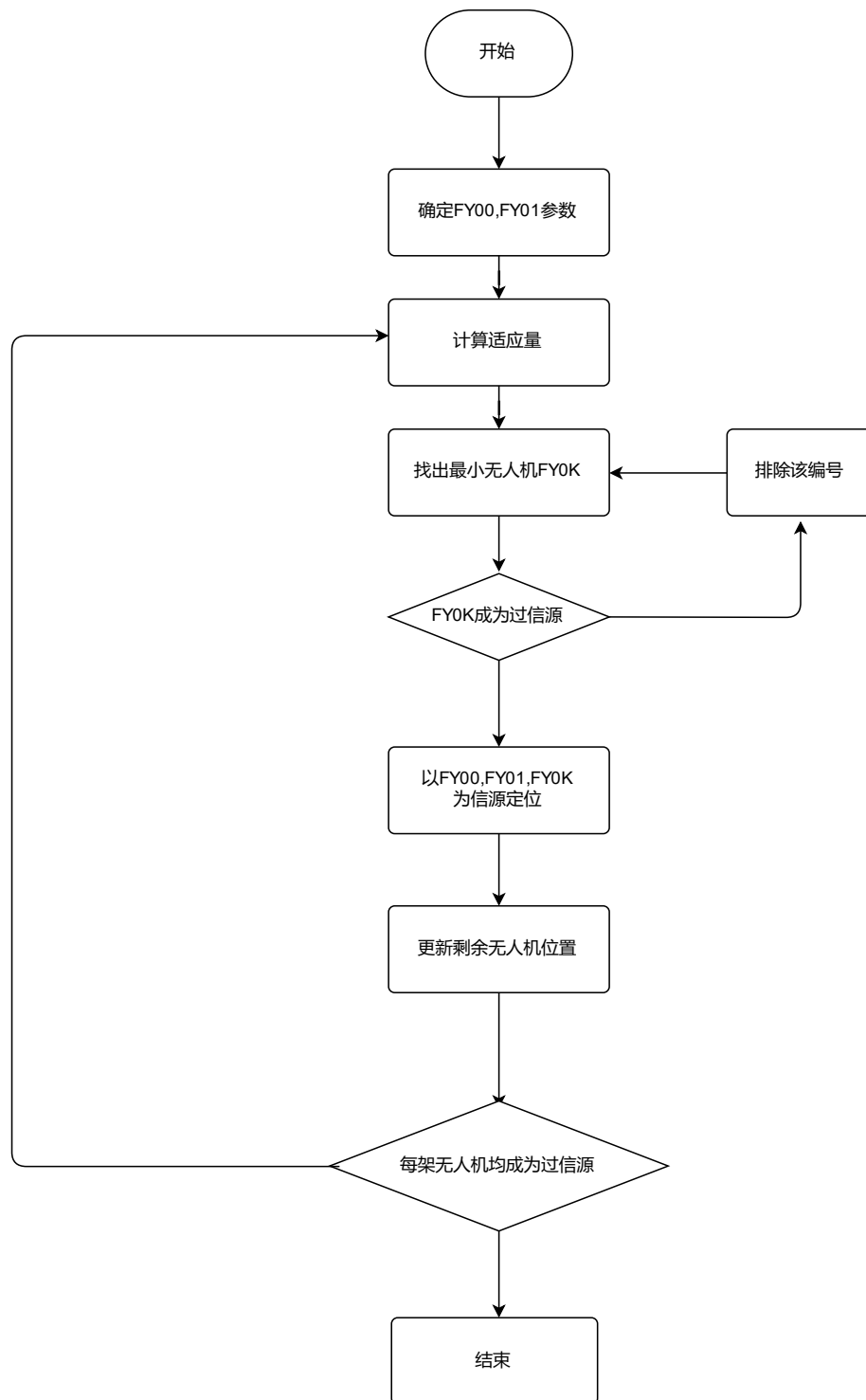


图 8

4.3.第三问

4.3.1 模型建立

题目给出了一系列存在定位偏差的无人机（其中只有 FY00 和 FY01 是准确的），本文中考虑采用每次选出一个位置最精确的无人机作为理想发射站进行迭代的方式进行定位更新。通过粒子群算法进行迭代迭代处理的流程如下所示：



4.3.2 模型求解

仿真结果如图所示。

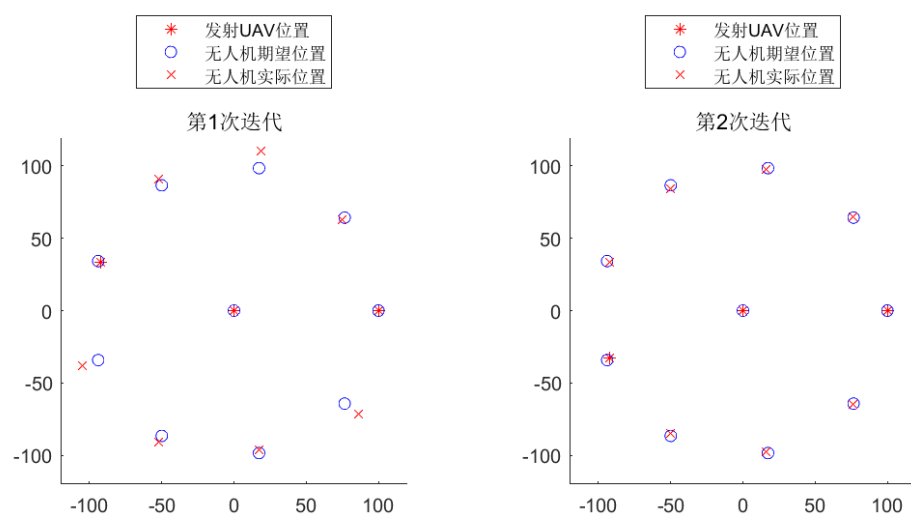


图 10

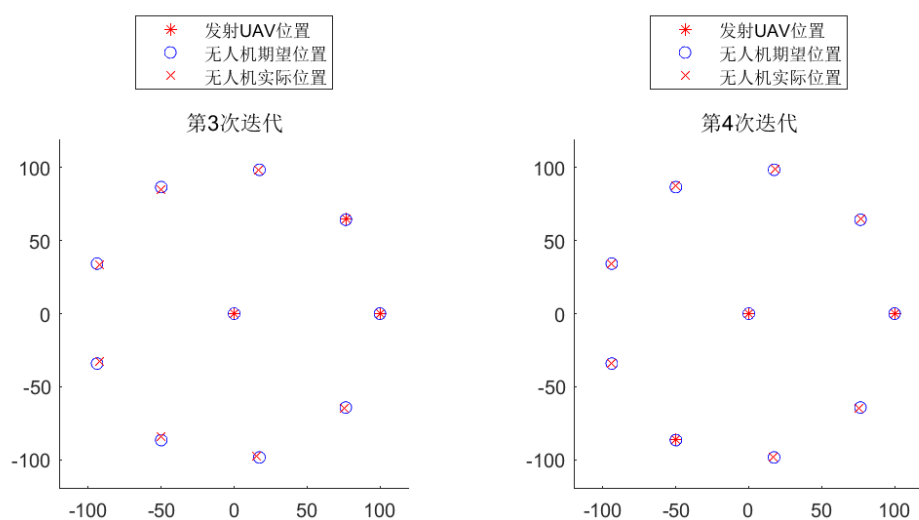


图 11

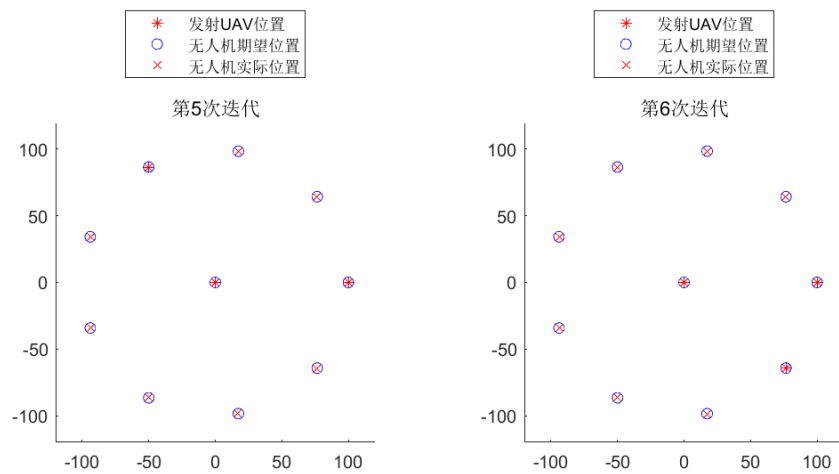


图 12

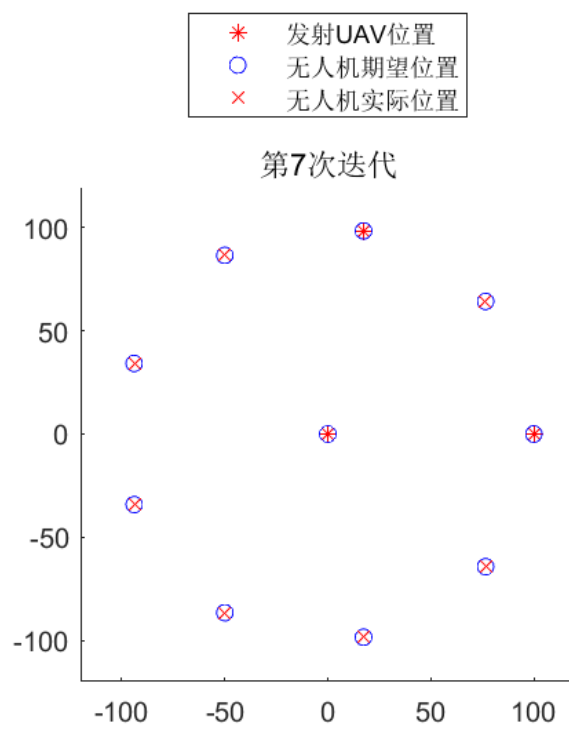


图 13

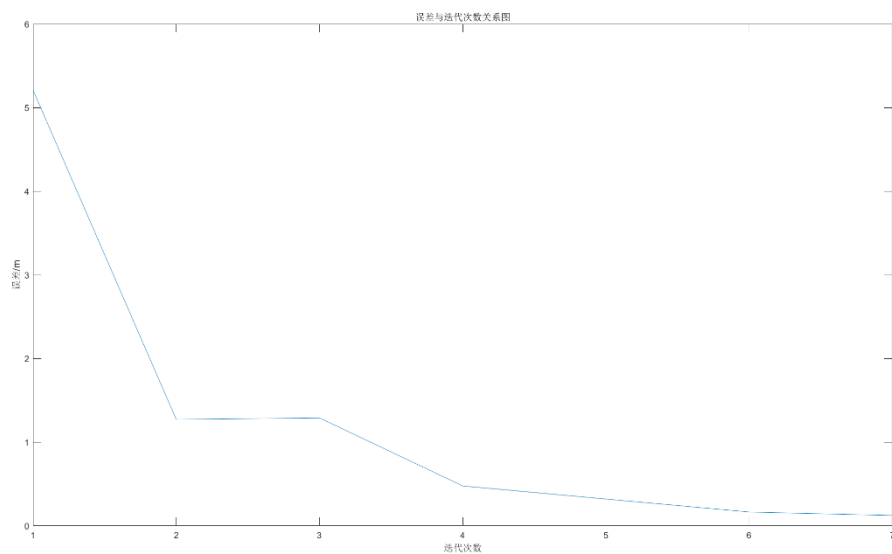


图 14

目标位置随着迭代的进行逐渐收敛。

4.3.3 调整方案

第 1 次调整,已校准无人机[FY0,FY1,FY5]

待校准无人机[FY6,FY8,FY7,FY4,FY2,FY3,FY9]

调整前误差 5.208263m,调整后误差 1.270646m

第 2 次调整,已校准无人机[FY0,FY1,FY6]

待校准无人机[FY9,FY2,FY8,FY7,FY4,FY3,FY5]

调整前误差 1.270646m,调整后误差 1.287183m

第 3 次调整,已校准无人机[FY0,FY1,FY2]

待校准无人机[FY3,FY4,FY7,FY9,FY8,FY5,FY6]

调整前误差 1.287183m,调整后误差 0.444161m

第 4 次调整,已校准无人机[FY0,FY1,FY7]

待校准无人机[FY4,FY8,FY3,FY9,FY2,FY5,FY6]

调整前误差 0.444161m,调整后误差 0.305233m

第 5 次调整,已校准无人机[FY0,FY1,FY4]

待校准无人机[FY3,FY8,FY9,FY2,FY5,FY6,FY7]

调整前误差 0.305233m,调整后误差 0.155501m

第 6 次调整,已校准无人机[FY0,FY1,FY8]

待校准无人机[FY9,FY3,FY2,FY4,FY5,FY6,FY7]

调整前误差 0.155501m,调整后误差 0.252065m

第 7 次调整,已校准无人机[FY0,FY1,FY3]

待校准无人机[FY9,FY2,FY4,FY5,FY6,FY7,FY8]

调整前误差 0.252065m,调整后误差 0.255706m

由上述数据可知，调整 6 次即可实现准确定位

4.4 模型评价

第一问依据几何关系，利用正弦定理表示出无人机在极坐标下的定位，同时对有偏差无人机的偏差范围进行了说明，同时表明了该模型成立的条件；第二问在第一问的基础上，采用数学方法证明了至少需要三架无人机即可精准定位；第三问在前两问的基础上，采用粒子群算法思想，在每个局部最优解种进行迭代以获得全局最优解，即准确定位。

5.问题二

5.1 模型建立

在问题二中无人机初始时刻坐标未知，当其编队队形为锥形时，仍利用无源定位的位置信息对队形进行调整。在该问题中，由于初始位置未知，我们无法直接通过无源定位的未知信息测量出每个无人机的位置。因此，需要通过相邻无人机之间距离以及角度来衡量无人机的位置偏差以及编队中存在的对称关系，对无人机的位置进行调整。我们可以通过贪婪策略，每次尽可能少的选择发射信号的无人机个数对其余无人机位置进行调整，以减少无人机向外发射电磁波信号，为产生干扰。

贪心算法指在对问题求解时，总是做出在当前看来是最好的选择。贪心选择是指所求问题的整体最优解可以通过一系列局部最优的选择，即贪心选择来达到。这是贪心算法可行的第一个基本要素。当一个问题最优解包含其子问题的最优解时，称此问题具有最优子结构性质。运用贪心策略在每一次转化时都取得了最优解。问选取应该满足局部优化的

条件。若下一个数据和部分最优解连在一起不再是可行解时，就不把该数据添加到部分解中，直到把所有数据枚举完，或者不能再添加算法停止。

在本问中，设置初始点为 FY01，设置选择下一个点进行修正的贪婪准则为：

1. 对点位置进行调整时，所需要的发射信号的无人机个数尽可能的少
2. 调整后的点能够尽可能的接近无人机群在实现锥形编队时的位置
3. 终止条件为对所有无人机位置均完成了调整

5.2 调整方案

基于上述贪婪准则，我们可以设计无人机位置的调整方案。

1、不妨设基准点 FY01,且 FY01 与 FY02 间距恰好为 50m，即 FY02 位置未偏离，则由前文可知，选取 FY01,FY02 为信源，通过有限次调整，使得 FY03 与 FY01、FY02 张角均为 60° ，此时可获得基准三角形 FY01-FY02-FY03，如下图所示

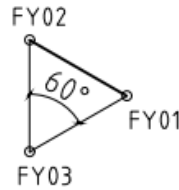


图 15

2、获得基准三角形后，为使得接下来所有调整的无人机均在同一参考平面内，我们需要从 FY01 相邻无人机中选择下一步进行调整的无人机，根据提出的贪婪策略，选择 FY05 进行调整，利用前文所提 AOA 三角定位法进行定位。

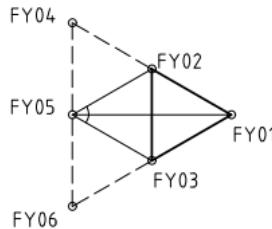


图 16

3、无人机 FY05 进行调整之后，我们从其相邻点中根据贪婪策略选取下次进行调整的无人机。由原队列对称性可知，此时 FY04 与 FY06 在进行贪婪选择时地位相同，不妨先调

整 FY04,使 FY01, FY02 发射的信号将 FY01，FY04 调整到同一条直线上，即 FY04 移动到 FY02-FY03-FY05 决定平面内。进而进行角度调整，使得 FY-04 对 FY-02、FY05 张角均为 60° 。

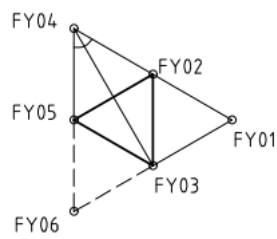


图 17

4、根据无人机 FY01, FY03 发射的信号将 FY01，FY06 调整到同一条直线上。再通过 FY02，FY03，FY05，即 FY06 移动到 FY02-FY03-FY05 决定平面内。进而进行角度调整，使得 FY-06 对 FY-03、FY05 张角均为 60°

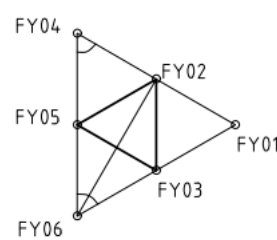


图 18

自此，我们将编号为 FY01-FY06 的无人机调整为了直线上相邻两机无人机的同一平面上间距相等的锥形队形。通过类似方法，我们可对剩余无人位置进行调整，最终可以将无人机 FY01-FY15 调整为锥形队形，并且满足直线上相邻两架无人机的间距相等。

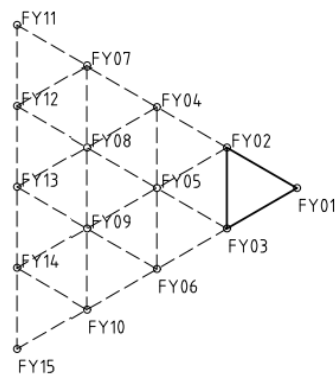


图 19

5.3 模型求解

5.3.1 数据处理：

不妨设 FY01,FY02,FY03 已到达期望位置，剩余无人机与期望位置间均存在误差，且误差符合 0-1 标准正态分布，利用 matlab 随机数生成可获得每架无人机的概率误差坐标 $[X_i,Y_i]$

5.3.2 模型求解

篇幅有限，此处仅展示一次生成结果

1、生成期望队列阵型

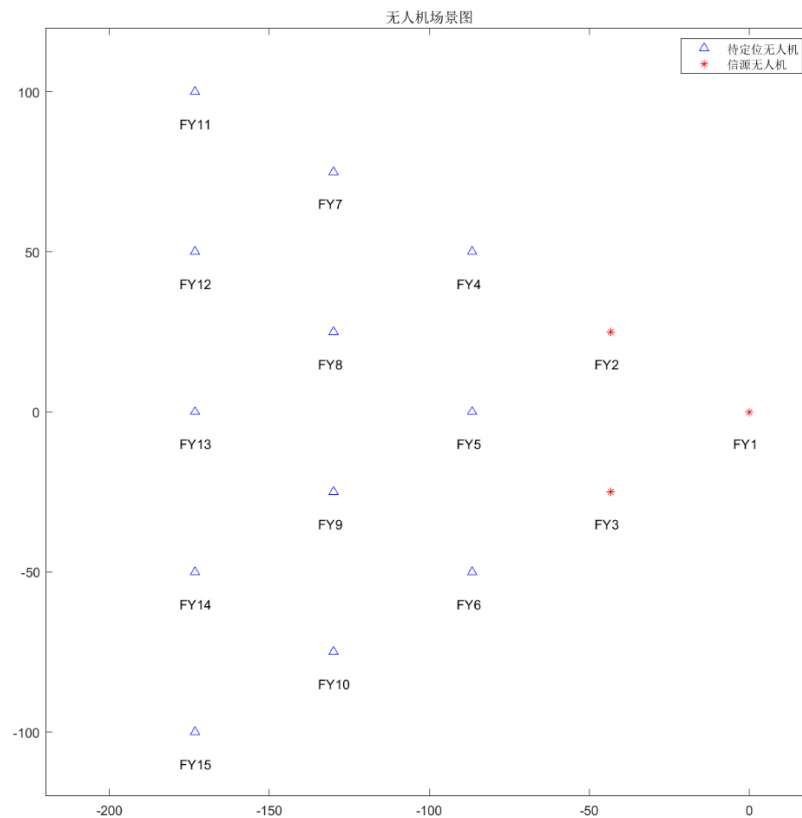


图 20

2、随机生成误差坐标

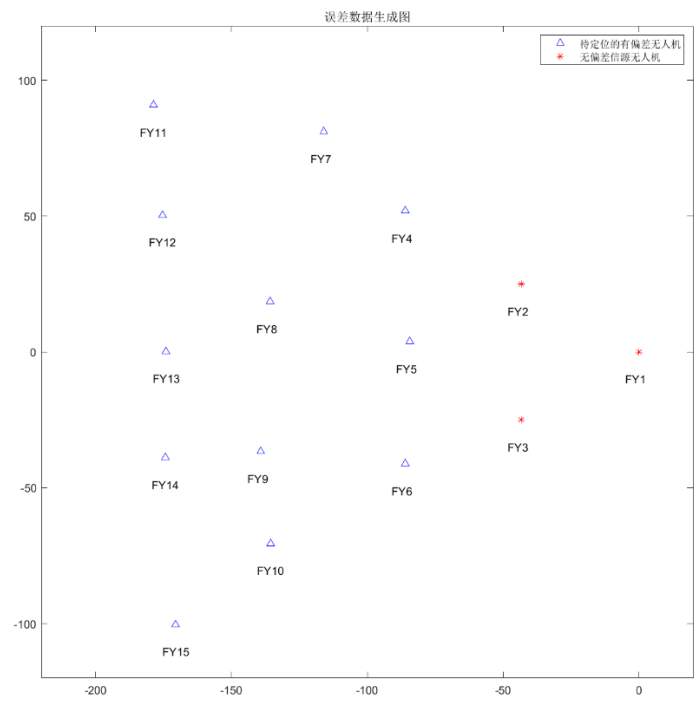


图 21

3、利用贪婪算法进行迭代

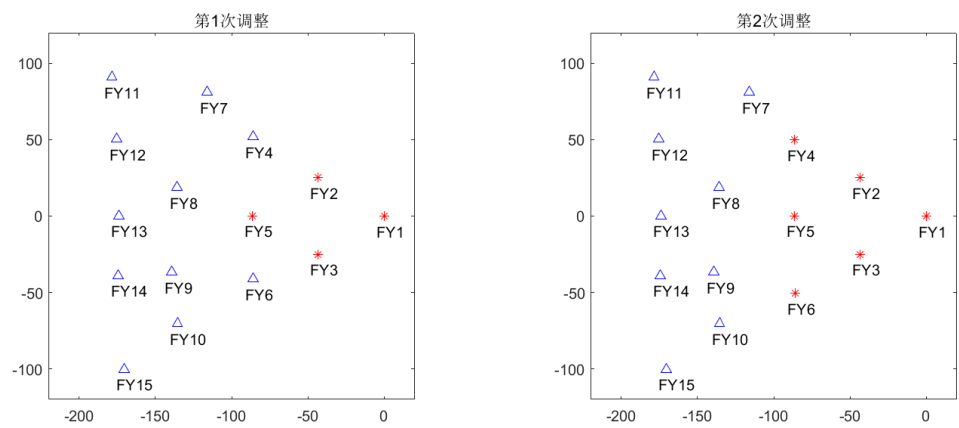


图 22

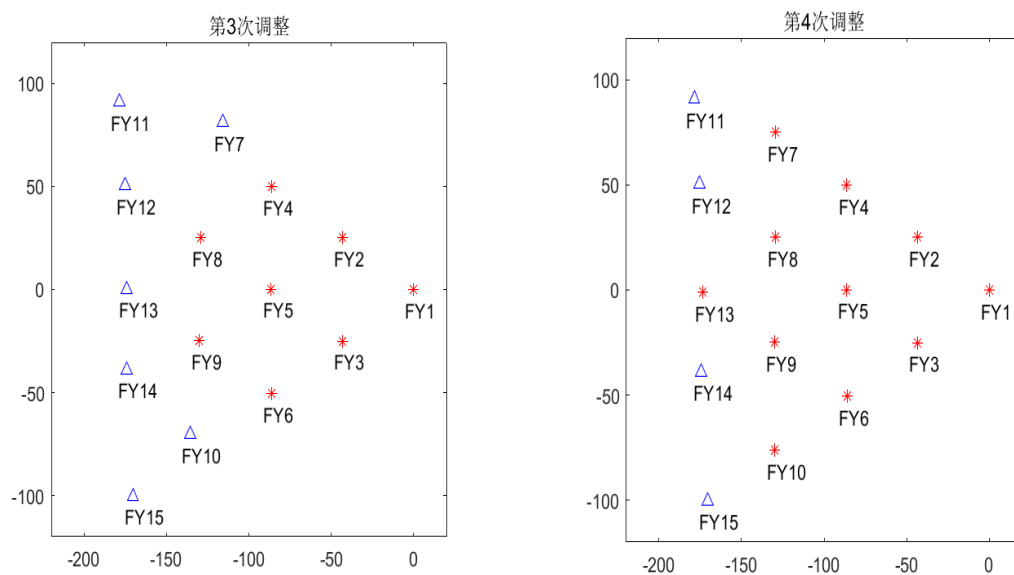


图 23

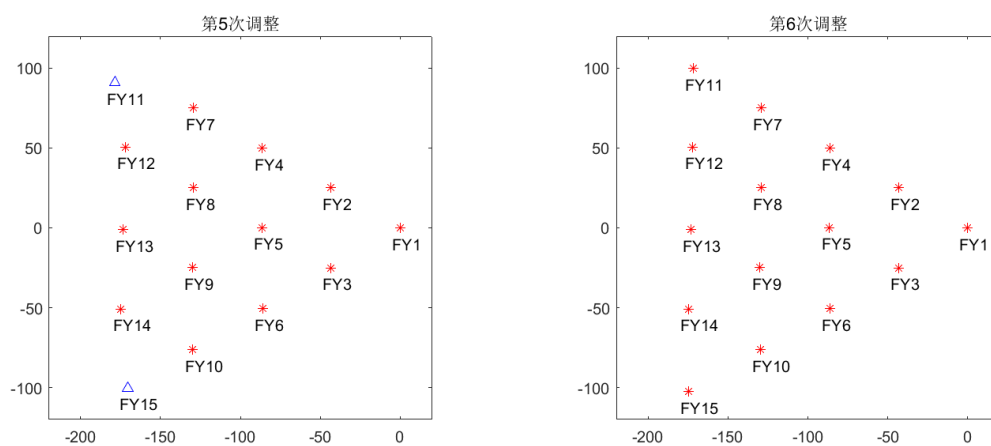


图 24

4、输出结果与期望值及原位置进行可视化比较

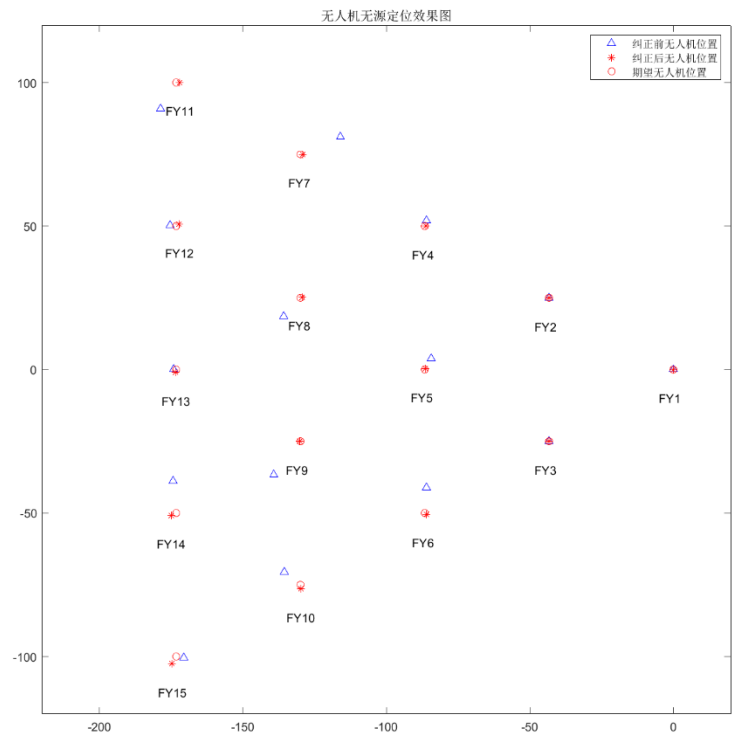


图 25

5、误差与迭代次数关系可视化

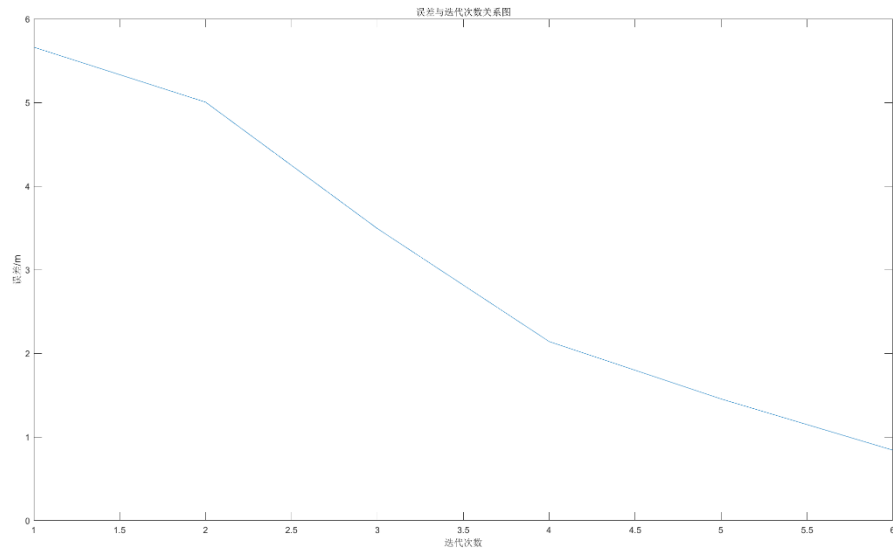


图 26

5.4 模型评价

如图所示，整个无人机机组完成了一次无人机位置更新和队形纠正。多次实验数据表明，按本文算法进行迭代 6 次后，无人机位置误差不超过 0.5m，即相对误差小于 1%。查阅相关资料可知，此时精度较高，模型拟合较为准确。

6 模型的优缺点及改进方向

模型优点：

1. 对于问题一的几何模型，简单直观，易于理解，且 AOA 三角定位充分利用无源定位所得的角度信息，易于在实际飞行测试中推广
2. 对于问题一的算法模型，采用 PSO 算法，步骤简单，收敛速度快，可调参数少，结果较为精确
3. 对于问题二，采用贪婪算法，并充分利用编队几何特性进行分析。检验模型时误差视为正态分布，更加贴合实际应用情景，可推广性强。
4. 在进行第二问求解时，将第一问所用的粒子群算法简化为贪婪算法，仅通过求局部最优解以得出结论，进一步简化计算及操作难度。

模型缺点：

1. AOA 三角定位算法需要配置复杂的天线系统，且角度误差对定位精度的影响远比测距误差大，算法精确度低。
2. 第一问在利用粒子群算法求调整方案时，若遇到特殊情况，小概率造成局部最优的情况
3. 第二问模型仅能解决二维编队问题，在处理大量无人机构成的复杂编队时无法保证所得解为全局最优

模型推广：

本文的编队方式默认所有的无人机都在同一二维平面上，但实际中无人机可能偏离预期高度。在这种情况下，根据空间 AOA 定位的原理，需要额外加入一架校准无人机，并通过重心定位法来降低 AOA 算法的敏感性，将内角和边长信息作为加权因子，以提高算法的精确度。同时给定约束条件，建立合适的优先级，利用离散粒子群算法(DPSO)进行迭代、寻找全局最优解。进行此改进后，模型将更加合理且完善，更有普适性。

参考文献

- [1]. 秦明星,王忠,李海龙,刘鑫.基于分布式模型预测的无人机编队避障控制[J/OL].北京航空航天大学学报:1-19[2022-09-17].
- [2]. 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.
- [3]. 丁兰兰,刘国栋.搭载 D-RTK GNSS 的大疆 M600 无人机定位分析与误差改正[B].地理空间信息, 1672-4623 (2020) 03-0016-04

附录

1.问题一第三问模型

算法 1:

```
clear all;clc;close all;

%% 设定参数
%无人机阵列在笛卡尔坐标系下的期望位置
R = 100;
expLoc = [0 0 0;
1 R 0;
2, R*cos(2/9*pi*1), R*sin(2/9*pi*1);
3, R*cos(2/9*pi*2), R*sin(2/9*pi*2);
4, R*cos(2/9*pi*3), R*sin(2/9*pi*3);
5, R*cos(2/9*pi*4), R*sin(2/9*pi*4);
6, R*cos(2/9*pi*5), R*sin(2/9*pi*5);
7, R*cos(2/9*pi*6), R*sin(2/9*pi*6);
8, R*cos(2/9*pi*7), R*sin(2/9*pi*7);
9, R*cos(2/9*pi*8), R*sin(2/9*pi*8)];

%实际目标位置(输入数据)
realLoc = [0 0 0;
1 100 0;
```

```

2, 98, 40.10
3, 112, 80.21
4, 105, 119.75
5, 98, 159.86
6, 112, 199.96
7, 105, 240.07
8, 98, 280.17
9, 112, 320.28];

for ii = 1:10
    realLocTemp(ii, :) = [ii - 1 [cosd(realLoc(ii, 3)) sind(realLoc(ii, 3))] * (realLoc(ii,
2))];
end

%% 排序
flag = [0 0 -1 -1 -1 -1 -1 -1 -1 -1];
dataseq(1).loc = realLocTemp;

for ii = 1:10
    dataseq(1).Err(ii) = norm(dataseq(1).loc(ii, 2:3) - expLoc(ii, 2:3));
end

maxIter = 8;
figure(1); clf;
IndSeqOld = [];

for it = 2:maxIter

    for ii = setdiff(3:10, IndSeqOld)
        angle_1_T = angCal(expLoc(1, 2:3), expLoc(ii, 2:3), expLoc(2, 2:3));
        angle_1 = angCal(dataseq(it - 1).loc(1, 2:3), dataseq(it - 1).loc(ii, 2:3),
        dataseq(it - 1).loc(2, 2:3));
        angle_2_T = angCal(expLoc(2, 2:3), expLoc(ii, 2:3), expLoc(1, 2:3));

```

```

angle_2 = angCal(dataseq(it - 1).loc(2, 2:3), dataseq(it - 1).loc(ii, 2:3),
dataseq(it - 1).loc(1, 2:3));
flag(ii) = abs(angle_1_T - angle_1)^2 + abs(angle_2_T - angle_2)^2;
end

[~, Ind] = sort(flag);
IndSeqOld = [IndSeqOld Ind(3)];
flag(Ind(3)) = 1000;
dataseq(it - 1).bsseq = [Ind(1:3)];
dataseq(it - 1).adjustseq = Ind(4:end);
subplot(2, 4, it - 1); hold on;
plot(dataseq(it - 1).loc(Ind(1:3), 2), dataseq(it - 1).loc(Ind(1:3), 3), 'r*');
plot(expLoc(:, 2), expLoc(:, 3), 'bo');
plot(dataseq(it - 1).loc(Ind(4:end), 2), dataseq(it - 1).loc(Ind(4:end), 3), 'rx');
axis square;
axis([-120 120 -120 120]);
legend('信源无人机位置','无人机期望位置','无人机实际位置
','Location='northoutside');
title(['第' num2str(it-1) '次迭代']);
fprintf('\n\n-----\n');
fprintf('第%d 次调整,已校准无人机[FY%d,FY%d,FY%d]\n',it-1,Ind(1)-1,Ind(2)-
1,Ind(3)-1);
fprintf('待校准无人机[FY%d,FY%d,FY%d,FY%d,FY%d,FY%d,FY%d]\n',Ind(4)-
1,Ind(5)-1,Ind(6)-1,Ind(7)-1,Ind(8)-1,Ind(9)-1,Ind(10)-1);
realLocTemp1=dataseq(it-1).loc;
Rxld = dataseq(it - 1).adjustseq;

for ic = 1:7
AngD = angleCal(realLocTemp1(Rxld(ic), 2:3), [Ind(1:3)], realLocTemp1(:,
2:3));
Loc = locFun(AngD, expLoc(:, 2:3), [Ind(3)], expLoc(Rxld(ic), 2:3));
moveDes = expLoc(Rxld(ic), 2:3) - Loc;
realLocTemp1(Rxld(ic), 2:3) = realLocTemp1(Rxld(ic), 2:3) + moveDes;

```

```
end

dataseq(it).loc = realLocTemp1;

for ii = 1:10
    dataseq(it).Err(ii) = norm(dataseq(it).loc(ii, 2:3) - expLoc(ii, 2:3));
end

fprintf('调整前误差%fm,调整后误差%fm\n', mean(dataseq(it - 1).Err),
        mean(dataseq(it).Err));

end

savefig('Pic-1.3.1.fig');

for it = 1:7
    res(it) = mean(dataseq(it).Err);
end

figure(2);
plot(res);
title('误差与迭代次数关系图')
xlabel('迭代次数');
ylabel('误差/m');
savefig('Pic-1.3.2.fig');
```

2.问题二模型

算法 2

```
clear all;clc;close all;
```

```

%% 参数设定
%无人机阵列在笛卡尔坐标系下的期望位置
R = 50;
X0 = R * cos(pi/6);
Y0 = R * sin(pi/6);
expLoc = [
    1 0 0;
    2 -X0 Y0;
    3, -X0, -Y0;
    4, -X0 * 2, Y0 * 2;
    5, -X0 * 2, 0;
    6, -X0 * 2, -Y0 * 2;
    7, -X0 * 3, Y0 * 3;
    8, -X0 * 3, Y0 * 1;
    9, -X0 * 3, -Y0 * 1;
    10, -X0 * 3, -Y0 * 3;
    11, -X0 * 4, Y0 * 4;
    12, -X0 * 4, Y0 * 2;
    13, -X0 * 4, Y0 * 0;
    14, -X0 * 4, -Y0 * 2;
    15, -X0 * 4, -Y0 * 4; ];
%基准点选择
benchMark = [1 2 3];
figure(1); clf; hold on; box on;
plot(expLoc(4:end, 2), expLoc(4:end, 3), 'b^');
plot(expLoc(benchMark, 2), expLoc(benchMark, 3), 'r*');

for ii = 1:15
    text(expLoc(ii, 2) - 5, expLoc(ii, 3) - 10, ['FY' num2str(ii)]);
end

axis square;
axis([-220 20 -120 120])

```

```

legend('待定位无人机','信源无人机')
title('无人机场景图');
savefig('Pic-2.1.fig');

%% 误差数据按 0-1 标准正太分布随机生成对应概率坐标
noiseSigma = 5;
expLocR = expLoc;
expLocR(4:end, 2:3) = expLocR(4:end, 2:3) + normrnd(0,1,[12 2]) * noiseSigma;
figure(2); clf; hold on; box on;
plot(expLocR(4:end, 2), expLocR(4:end, 3), 'b^');
plot(expLocR(benchMark, 2), expLocR(benchMark, 3), 'r*');

for ii = 1:15
    text(expLocR(ii, 2) - 5, expLocR(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','无偏差信源无人机')
title('误差数据生成图')
savefig('Pic-2.2.fig');
expLocR2 = expLocR;
%%定义父坐标
uavFatherId = [
    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; ];
%%判断需要调整的无人机编号
uavToAdjust = [0 0 0 1 1 1 1 1 1 1 1 1 1 1 1];
figure(3); clf;
count = 1;

while (sum(uavToAdjust) > 0)
    Ind = find(uavToAdjust == 1);
    temp = uavToAdjust;
    % 控制台输出调整情况
    fprintf('\n\n-----\n');
    fprintf('第%d 次调整,已校准无人机[,count);

    for ii = 1:15

        if (uavToAdjust(ii) == 0)
            fprintf('FY%d,', ii);
        end

    end

    fprintf('\b]\n 待调整无人机[');

    adjustIter = [];

    for ii = Ind

        if (sum(uavToAdjust(uavFatherId(ii, 1:3))) == 0)
            ang = angCal(expLocR(ii, 2:3), uavFatherId(ii, 1:3), expLoc(:, 2:3));
            Loc = locFun(ang, uavFatherId(ii, 1:3), expLoc(:, 2:3), uavFatherId(ii, 4));

```

```

        moveDirction = expLoc(ii, 2:3) - Loc;
        expLocR(ii, 2:3) = expLocR(ii, 2:3) + moveDirction;
        temp(ii) = 0;
        adjustlter = [adjustlter ii];
    end

end

uavToAdjust = temp;

for ii = 1:15
    dataseq(count).Err(ii) = norm(expLocR(ii, 2:3) - expLoc(ii, 2:3));
end

for ii = adjustlter
    fprintf('FY%d,', ii);
end

fprintf('\b]\n 调整后误差%fm\n',mean(dataseq(count).Err));

subplot(2, 3, count); hold on; box on;
plot(expLocR(uavToAdjust == 1, 2), expLocR(uavToAdjust == 1, 3), 'b^');
plot(expLocR(uavToAdjust == 0, 2), expLocR(uavToAdjust == 0, 3), 'r*');

for ii = 1:15
    text(expLocR(ii, 2) - 5, expLocR(ii, 3) - 10, ['FY' num2str(ii)]);
end

axis square;
axis([-220 20 -120 120])
title(['第' num2str(count) '次调整']);

count = count + 1;

end

```

```
savefig('Pic-5.3.fig');
```

```
for it = 1:count - 1  
    res(it) = mean(dataseq(it).Err);  
end
```

```
figure(4);  
plot(res);  
xlabel('迭代次数');  
ylabel('误差/m');  
title('误差与迭代次数关系图')  
savefig('Pic-2.4.fig');
```

```
figure(5); clf; hold on; box on;  
plot(expLocR2(:, 2), expLocR2(:, 3), 'b^');  
plot(expLocR(:, 2), expLocR(:, 3), 'r*');  
plot(expLoc(:, 2), expLoc(:, 3), 'ro');
```

```
for ii = 1:15  
    text(expLocR(ii, 2) - 5, expLocR(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')  
title('无人机无源定位效果图')  
savefig('Pic-2.5.fig');
```
