

基于仿真模拟的三角形纯方位无源定位模型

摘要

本文主要通过建立三角形纯方位无源定位模型来确定圆形编队中无人机的位置，通过仿真模拟检验模型的实际意义，并将模型应用于不同的编队中。

针对问题 1(1)，在实际生活中，角度测量都会存在一定的误差，为了确定接收信号无人机的位置，建立三角形纯方位无源定位模型。假设角度测量呈正态分布，则测量点会存在于某个定位三角形中，为找出该三角形中可以代替测量位置的点，引入重心和内心。仿真模拟出 10000 个正态分布生成的点，分别与内心和重心求平均距离。模拟结果显示，与内心的平均距离为 0.394，与重心的平均距离为 0.403，即用内心代替测量点实际位置更好。

针对问题 1(2)，由问题一得出，三架无人机形成的定位三角形内心与测量点之间仍然有一定的距离。用内心代替测量点，存在偶然性，不能有效定位。在此基础上，建立四点定位模型：四架无人机发射信号，可以形成 4 个定位三角形，测量点处在 4 个定位三角形相交的公共区域中，面积远小于任一三角形面积，减少误差。取四个定位三角形的内心坐标平均值为无人机实际的位置，得以有效定位。故除 FY00 和 FY01 外，还需要两架无人机可以实现其余无人机有效定位。

针对问题 1(3)，由题目所给数据和四点定位模型，每次调整无人机位置时，由 FY00, FY01 和圆周上的两个点发射信号，确定并调整其余无人机的位置。经过多次调整，使得最终所有外围无人机都落在半径为 100，两两相隔 40 度的圆周上。角度调整用时约为 2s，距离调整用时约 3s。

针对问题 2，根据题 1(2) 的结论，四点定位模型可以有效定位测量点。为探究锥形编队调整的普遍方案，我们探究由 6 架无人机组成的锥形编队。固定锥尖无人机位置，每次调整通过其与另外三架无人机发射信号，其余接收信号，调整位置。经过多次调整，使所有无人机都落在锥形编队上。角度用时约 5s，距离调整用时约 3s。

关键字： 无人机 三角形纯方位无源定位模型 仿真模拟 内心 定位三角形 圆形编队 锥形编队

一、问题重述

1.1 背景知识

无人机在当今的民用和军用方面都有很大的作用，民用主要用于航拍，农业，测绘等领域，军用主要用于侦察，拦截目标等领域。在军用中，无人机常以编队的形式执行任务，为避免被外界干扰以及被雷达探测，应当减少电磁信号的传播。如何在不同编队队形中，通过无源探测定位的方式来有效调整无人机的位置显得尤为重要。

1.2 要解决的问题

根据题目所给条件及数据，我们应解决以下四个问题：

问题一：在圆形编队中，已知编号且位置准确，位于圆心和圆周上共计三架无人机发射信号，如何通过接收信息，建立起定位模型，分析其余无人机的定位。

问题二：在圆形编队中，已知圆心和圆周上各一架无人机的准确定位，求还需要的拥有准确定位的无人机数量，通过这些无人机发射信息，使得剩余无人机可以被有效定位。

问题三：在圆形编队中，已知 10 架无人机的初始定位，设计无人机调整方案，每次通过 FY00 和圆周上的最多三架无人机发射信号，其余无人机接收信号并调整位置，最终使 9 架无人机都在一确定圆周上均匀分布。

问题四：在其他编队队形中，运用纯方位无源定位，如何设计方案，调整无人机位置。

二、问题分析

2.1 问题 1(1) 的分析

如果无人机角度测量准确，那么三架无人机可以确定其余无人机的准确位置。但在实际的应用中，无人机角度测量会存在一定的误差，使得接收信息的无人机的位置存在于一个区域中，而不是具有确定坐标的某个点上。通过分析角度误差，我们建立出无人机区域定位模型。

2.2 问题 1(2) 的分析

由问题 1(1) 分析可知, 由于角度误差的存在, 无论多少架发射信号无人机也无法将接收信号的无人机确定在一个确定坐标的点。随着发射信息的无人机数量增多, 其最终所确定的区域面积将逐渐变小。而为了减少电磁波信号, 应当在提高定位精度的同时减少发射信号的无人机的数量。通过权衡二者之间的关系, 最终寻找出可以既可以有效定位, 又发射较少电磁波的无人机数量。

2.3 问题 1(3) 的分析

通过对第一第二题的分析, 将原本处于区域范围内的点定位于概率最大的位置上, 运用题目中所给出的坐标数据, 通过仿真模拟, 进行多次调整, 最终使各架无人机处于同一个圆周上, 并得到该调整方案的合理性和精确程度。

2.4 问题 2 的分析

通过前几题的分析, 固定锥形编队锥尖的位置, 编队中三架无人机与锥尖一起发射信号, 其余无人机接收信号。应用四点定位模型, 确定, 调整其余无人机的位置。多次改变编队中发射信号的无人机编号, 最后经过多次微调, 使无人机基本处于锥形编队之中。

三、 模型假设

- 测量过程中只存在角度测量误差, 不考虑包括电磁干扰等其他误差。
- 角度误差成正态分布。
- 有偏差的无人机偏差范围较小。
- 发射信号无人机位置不发生偏差。
- 无人机调整时间忽略不计。

四、符号说明

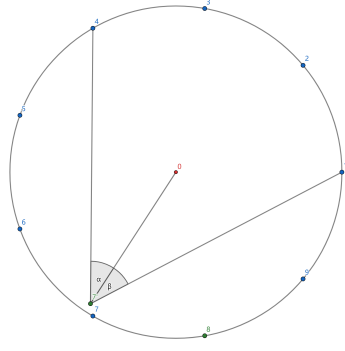
符号	意义
α_i	观测点与两发射点夹角
α	观测点与圆心连线和竖直方向的夹角
θ_i	观测点与圆周点连线和竖直方向的夹角
λ	误差参数
x_{ij}	发射线交点横坐标
y_{ij}	发射线交点纵坐标
(X_O, Y_O)	重心坐标
(X_I, Y_I)	内心坐标
\bar{x}	横坐标均值
\bar{y}	纵坐标均值
x_0	横坐标方差
y_0	纵坐标方差
d	点距

五、模型的建立与求解

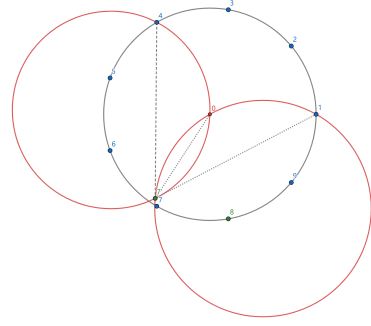
5.1 无人机定位模型的建立与求解

三点纯方位定位模型，是指三点同时向观测点发射信号，观测点接收到角度信号，若无角度接收误差，则通过几何关系可以直接确定观测点准确位置（图 1 (a)）。事实上，该点即为两个确定圆的交点（图 1 (b)）。但在实际情况下，一定会存在不可避免的测量误差。为此，我们引入一个误差参数 λ 来描述测量角度与真实值的差距。在此设：

$$\lambda \sim N(1, 1)$$

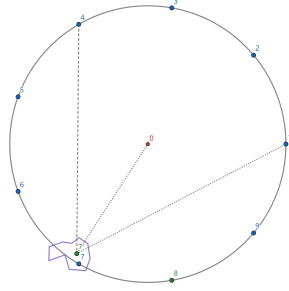


(a)

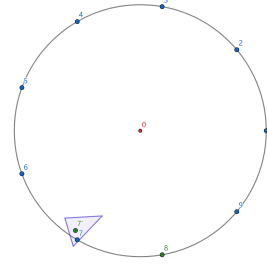


(b)

图 1



(a)



(b)

图 2

在此情况下， $\alpha' = \lambda\alpha$ 。因此，得到的无人机位置实际上会落在该平面上的一个区域内（图 2 (a)）。事实上，对于依靠两圆交点和误差参数得到的区域，其边界的求解是极为困难的。因此，在模型的建立中，我们进行了优化，使得该区域形成一个三角形（图 2 (b)）。进一步，通过仿真模拟，在发射角呈正态分布的条件下，通过计算所有点与三角形重心和内心的平均距离，最终确定以何点来确立观测点的实际位置。

5.1.1 计算角度关系

不失一般性，我们可以将圆心与圆周上的发射点分别记坐标为 $(0, 0)$, $(1, 0)$, $(\cos \frac{2k\pi}{9}, \sin \frac{2k\pi}{9})$ 。将他们分别与观测点相连，测得两两与观测点所成夹角角度分别为 $\alpha_1, \alpha_2, \alpha_1 + \alpha_2$

他们与观测点的的连线与竖直方向的夹角（图 3）分别设为 $\alpha, \theta_1, \theta_2$, 据此有方程组:

$$\begin{cases} \alpha + \theta_2 = \alpha_1 \\ \theta_1 - \theta_2 = \alpha_2 \end{cases}$$

由题所知，无人机位置略有偏差，于是可认为圆心到观测点对应的圆周点之间的连线

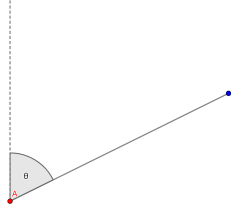


图 3 与测量点的连线同竖直方向形成的夹角示意图

与竖直方向的夹角和实际位置连线与竖直方向的夹角近乎相等，即 $\alpha = \alpha_{real}$ 。在此基础上， θ_1, θ_2 可通过方程组解出。有

$$\theta_1 = \alpha_1 + \alpha_2 - \alpha, \theta_2 = \alpha_1 - \alpha$$

由于存在观测误差，我们假定 α_1, α_2 成正态分布。自然的， θ_1, θ_2 也成正态分布，由此形成三角形定位区域（图 4），其实际上是两直线旋转过程中所有的交点形成的区域。

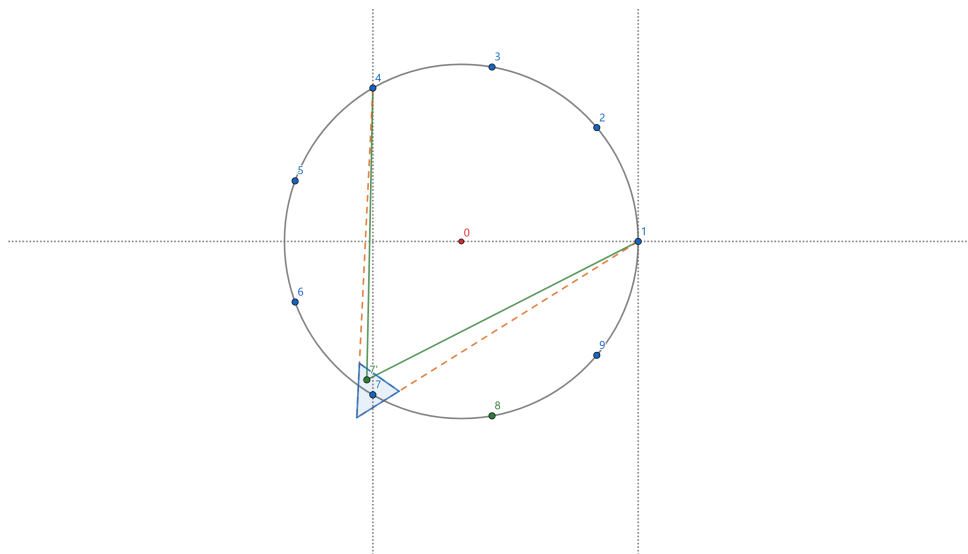


图 4 优化后定位法

通过计算，我们得到：

圆心发射线与圆周 1 点发射线的交点 (x_{12}, y_{12}) 为：

$$x_{12} = \frac{\sin \alpha \cos (\alpha_1 - \alpha)}{\sin (2\alpha - \alpha_1)}$$

$$y_{12} = \frac{\sin \alpha \sin (\alpha_1 - \alpha)}{\sin (2\alpha - \alpha_1)}$$

圆心发射线与圆周 2 点发射线的交点 (x_{13}, y_{13}) 为：

$$x_{13} = \frac{\cos (\frac{2k\pi}{9}) \sin (\alpha) \cos (\alpha_2 + \alpha_1 - \alpha) - \sin (\frac{2k\pi}{9}) \sin (\alpha) \sin (\alpha_2 + \alpha_1 - \alpha)}{\sin (2\alpha - \alpha_1)}$$

$$y_{13} = \frac{-\sin (\frac{2k\pi}{9}) \cos (\alpha) \cos (\alpha_2 + \alpha_1 - \alpha) + \cos (\frac{2k\pi}{9}) \sin (\alpha) \sin (\alpha_2 + \alpha_1 - \alpha)}{\sin (2\alpha - \alpha_1 - \alpha_2)}$$

圆周 1 点发射线与圆周 2 点发射线的交点 (x_{23}, y_{23}) 为：

$$x_{23} = \frac{\cos (\frac{2k\pi}{9}) \sin (\alpha_1 - \alpha) \cos (\alpha_2 + \alpha_1 - \alpha)}{\sin (2\alpha - \alpha_1)}$$

$$- \frac{\sin (\frac{2k\pi}{9}) \sin (\alpha_1 - \alpha) \sin (\alpha_2 + \alpha_1 - \alpha) - \cos (\alpha_1 - \alpha) \sin (\alpha_1 + \alpha_2 - \alpha)}{\sin (2\alpha - \alpha_1)}$$

$$y_{23} = \frac{\sin (\frac{2k\pi}{9}) \cos (\alpha_1 - \alpha) \sin (\alpha_2 + \alpha_1 - \alpha) + (1 - \cos (\frac{2k\pi}{9})) \sin (\alpha_1 - \alpha) \sin (\alpha_2 + \alpha_1 - \alpha)}{\sin (\alpha_2)}$$

5.1.2 两种定位点的计算方式

重心法：

$$X_I = \frac{x_{12} + x_{13} + x_{23}}{3}$$

$$Y_I = \frac{y_{12} + y_{13} + y_{23}}{3}$$

内心法：

$$X_O = \frac{Ax_{12} + Bx_{13} + Cx_{23}}{A + B + C}$$

$$Y_O = \frac{Ay_{12} + By_{13} + Cy_{23}}{A + B + C}$$

其中：

$$A = \sqrt{(x_{13} - x_{23})^2 + (y_{13} - y_{23})^2}$$

$$B = \sqrt{(x_{12} - x_{23})^2 + (y_{12} - y_{23})^2}$$

$$C = \sqrt{(x_{12} - x_{13})^2 + (y_{12} - y_{13})^2}$$

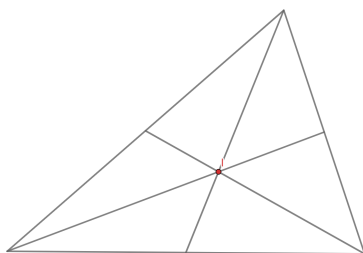


图5 重心

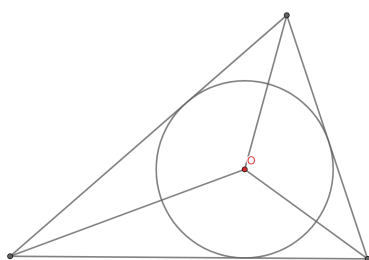


图6 内心

精确位置：

$$X = \frac{A_1 x_{12} + B_1 x_{13} + C_1 x_{23}}{4(A_1 + B_2 + C_3)} + \frac{A_2 x_{13} + B_2 x_{14} + C_2 x_{34}}{4(A_2 + B_2 + C_2)} \\ + \frac{A_3 x_{12} + B_3 x_{14} + C_3 x_{24}}{4(A_3 + B_3 + C_3)} + \frac{A_4 x_{23} + B_4 x_{24} + C_4 x_{34}}{4(A_4 + B_4 + C_4)}$$

$$Y = \frac{A_1 y_{12} + B_1 y_{13} + C_1 y_{23}}{4(A_1 + B_2 + C_3)} + \frac{A_2 y_{13} + B_2 y_{14} + C_2 y_{34}}{4(A_2 + B_2 + C_2)} \\ + \frac{A_3 y_{12} + B_3 y_{14} + C_3 y_{24}}{4(A_3 + B_3 + C_3)} + \frac{A_4 y_{23} + B_4 y_{24} + C_4 y_{34}}{4(A_4 + B_4 + C_4)}$$

5.1.3 两种方法的比较

通过三点纯方位三角定位，我们虽然可以算出定位三角形的坐标，但其是一个具有正态分布变量的复杂函数，直接计算较为困难。因此，我们引入 Monte Carlo 仿真模拟来比较两种方法。

1. 依正态分布 $\alpha_i N(\alpha, \sigma)$ 中产生 n 个数。（取 $n = 10000$ ）
2. 依公式依次算出第 i 次定位三角形的顶点坐标，记为 $(x_{12_i}, y_{12_i}), (x_{13_i}, y_{13_i}), (x_{23_i}, y_{23_i}), i = 1 \cdots n$
3. 根据步骤 2 的结果，分别求出第 i 次相应的重心与内心坐标，记为 $(x_{a_i}, y_{a_i}), (x_{b_i}, y_{b_i})$
4. 根据步骤三的结果，分别求取样本均值，方差，以重心方法为例，计算公式如下，样本均值为：

$$\bar{x} = \sum_{i=1}^n \frac{x_{a_i}}{n}$$
$$\bar{y} = \sum_{i=1}^n \frac{y_{a_i}}{n}$$

样本方差为：

$$x_0 = \sum_{i=1}^n \frac{(x_{a_i} - \bar{x})^2}{n}$$
$$y_0 = \sum_{i=1}^n \frac{(y_{a_i} - \bar{y})^2}{n}$$

另一方面，一个重要的衡量标准是依正态分布所得的各模拟点与内心及重心之间的距离：

$$d_I = \sqrt{(x_i - \bar{x}_a)^2 + (y_i - \bar{y}_a)^2}$$
$$d_O = \sqrt{(x_i - \bar{x}_b)^2 + (y_i - \bar{y}_b)^2}$$

所得值反应内心与重心同模拟点的贴近程度。

5. 通过分析以上数据，进而比较出两种方法中定位误差更小的一种。

5.1.4 比较结果分析

通过仿真模拟，我们得到了在内心与重心两种情况下， n 个模拟点对于其二者分别得距离之和（ $n = 10000$ ）并选取其中一百个点作出实际位置图片（图 7）由此可见，内心相较于重心更贴近模拟点，选取内心可以使定位误差更小，提高定位的精确度。

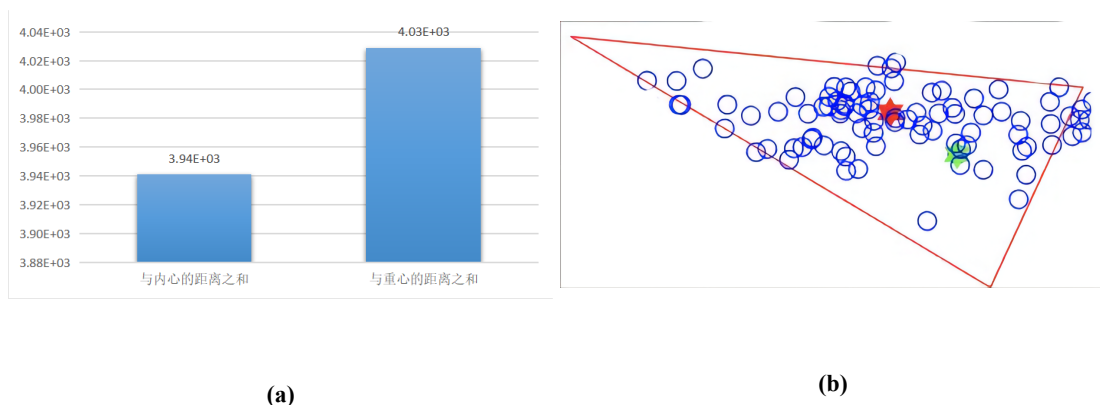


图 7 仿真模拟结果

5.2 判定无人机的有效定位架数

由第一题我们可以得到，在角度测量存在误差的情况下，无人机精确位置存在于定位三角形区域中，通过分析我们将定位三角形的内心定为无人机位置。但这种定位方式存在偶然性，可能会出现较大的误差，不符合有效定位。如果再增加一架发射信号的无人机，则四架无人机中的每三架可以形成一个定位三角形，一共形成了四个定位三角形以及相对应的四个定位内心。这四个定位三角形所围成的定位区域面积大幅度减小，基于此，我们取四个定位内心坐标的平均值即可实现对其余无人机的有效定位。

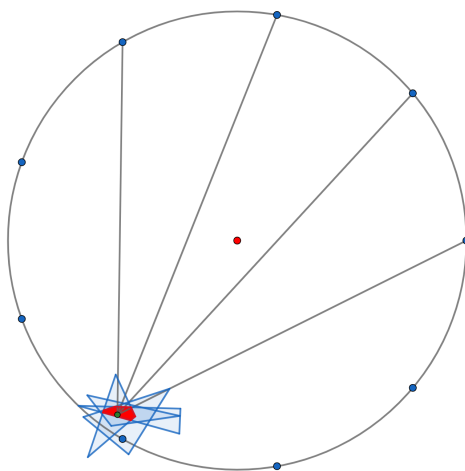


图 8 4 个定位三角形围成区域

5.3 无人机调整方案

5.3.1 方案设计

由前两题可知，通过圆心上的点以及圆周上的三个点发射信号，可以较为有效的定位其余飞机。通过数据分析，确定最终在圆周上的飞机应为半径为 100 米且两两间隔 40 度。因此，每次固定 **FY00** 与 **FY01** 发射信号。调整步骤如下：

- 1. 先使 **0, 1, 4, 7** 号发射信号，使其他无人机根据接收角度和利用四点三角形定位模型，确定自身位置。再向半径为 100，且角度符合的方向前进。例如：2 号飞机向极坐标下的点 $(100, 40^\circ)$ 的方向飞行。
- 2. 再使 **0, 1, 6, 9** 号发射信号，第一次调整的飞机位置微调，4, 7 号飞机位置向半径 100 的圆周靠近。例如：7 号飞机向极坐标下的点 $(100, 240^\circ)$ 的方向飞行。
- 3. 经过两次调整后，外围飞机基本都处在圆周上。再使 **0, 1, 2, 8** 号飞机发射信号，使得其余飞机位置微调。
- 4. 最终 1-9 号无人机都基本处在一个半径为 100 的圆周上。

5.3.2 方案合理性及精度检验

依靠此方案, 进行计算机仿真模拟调整，我们得到了调整结果如图 9。由此可见，在调整结束是，点的位置与预期位置差距极小。因此，该调整方案能实现预期目标，较为合理，且基于此调整方案的调整过程具有较高精度，定位效果良好。

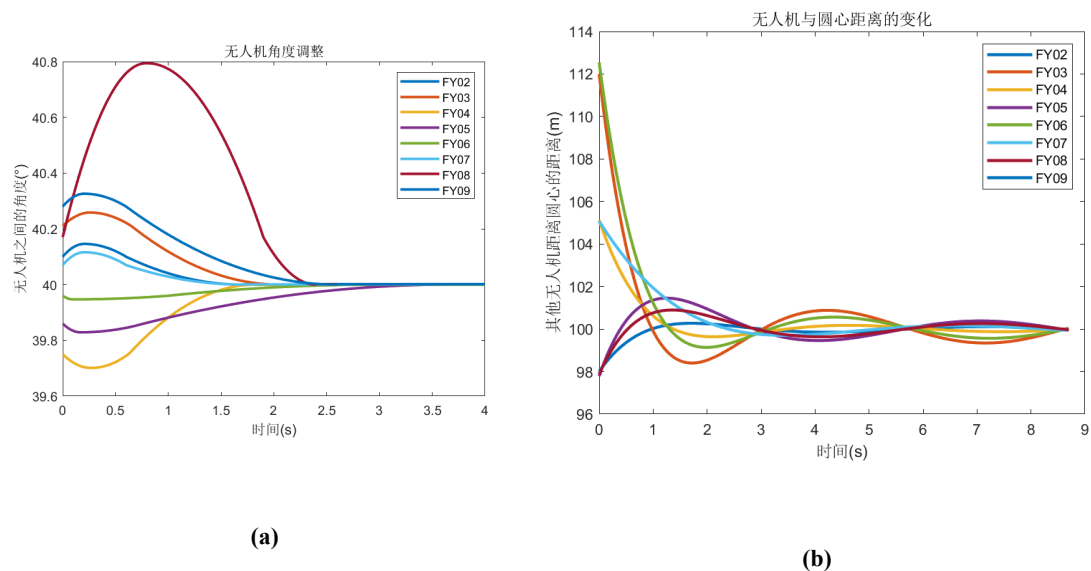


图 9

5.4 锥形编队的调整方案

5.4.1 方案设计

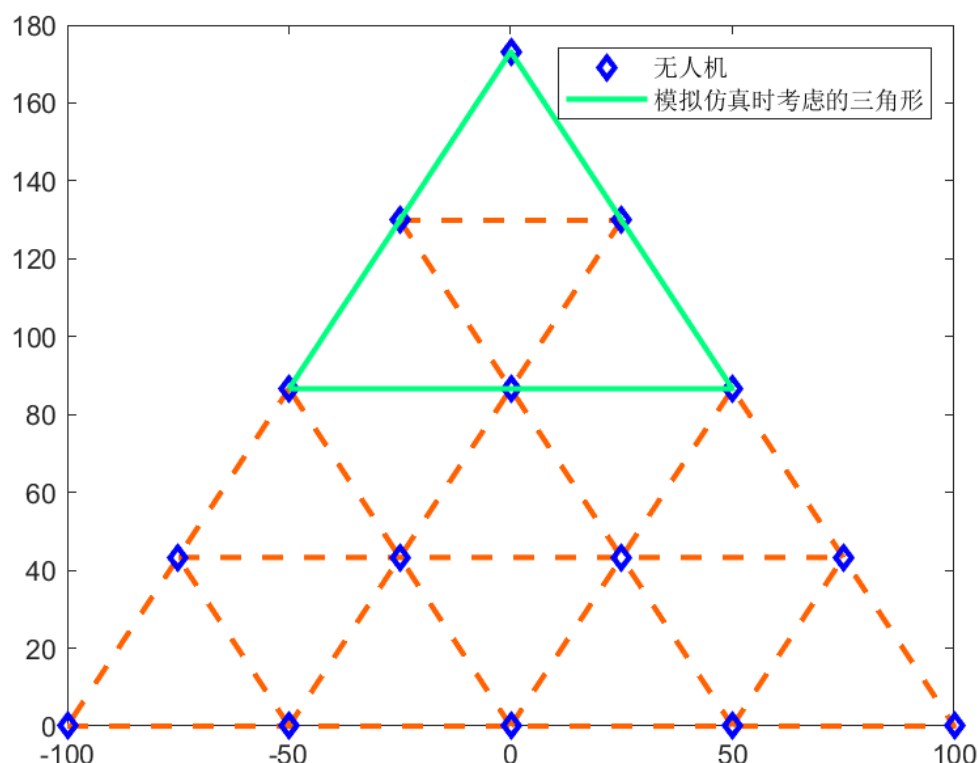


图 10 4 个定位三角形围成区域

由题 1 (2) 可知，四点发射信号，可以确定其余无人机的有效定位。为保证调整的有序性，每次固定锥形顶点（即 1 号无人机）的定位，并选出队列中三个无人机同时发出信号，使其余无人机接收信号并调整，最终使无人机落在两两间隔 50 米的锥形区域。为了仿真模拟的有效性，我们调整 6 架无人机组成的锥形队列。其余数量的锥形编队调整与之类似。调整步骤如下：

1. 1, 4, 5, 6 号无人机发射信号，2, 3 号无人机接收信号，根据四点定位模型判断定位，调整位置到锥形编队上。
2. 1, 2, 3, 6 号无人机发射信号，5 号接收定位信号，调整位置；4 号位置微调。
3. 1, 2, 3, 4 号无人机发射信号，5, 6 号接收信号，位置微调。
4. 1, 4, 5, 6 号无人机发射信号，2, 3 号接收信号，位置微调。
5. 最终使无人机落在如图锥形编队上。

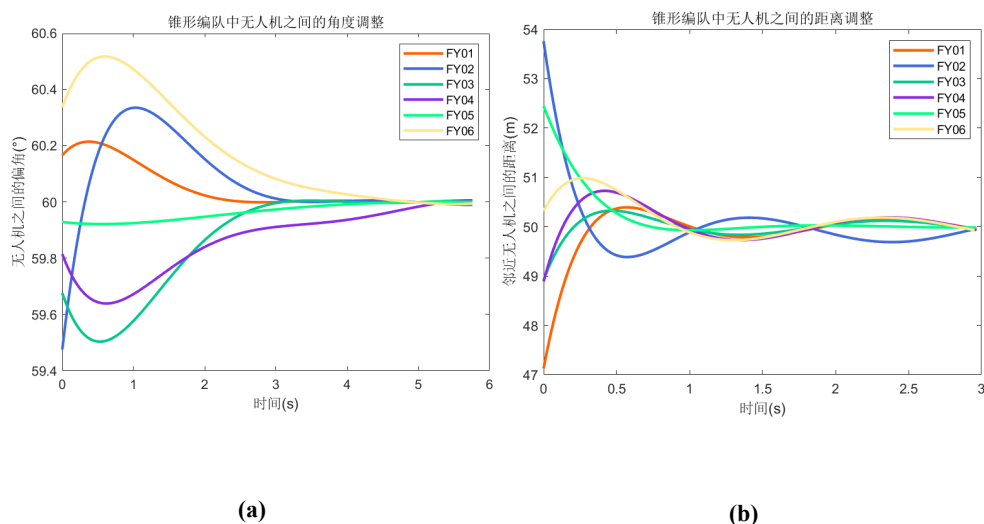


图 11

5.4.2 方案合理性及精度检验

由模拟调整结果（图 11）可见，该调整效果较为精确，与预期目标较为吻合，因此具有合理性。

六、 模型的评价及推广

6.1 模型的优点

三点定位三角形模型考虑了无人机角度测量的误差情况，并假设角度误差呈正态分布，符合实际情况。通过建立出无人机的定位三角形，找出其内心代替测量点的位置，减小误差。为减少测量的偶然性，在该模型上，引入的四点定位区域模型，大大减小了定位范围，取四个定位三角形内心的平均坐标，使得测量出的结果与实际更加符合，从而可以有效提高定位精度。在实际应用中，通过仿真模拟与实际数据，该模型定位与实际基本吻合，具有实际意义，可用于无人机编队的调整，更能广泛适用于各种不同类型的无人机编队。

6.2 模型的缺点

未运用更多的模型与三角形定位模型相结合，使得定位进一步精准。对于角度误差的补偿算法可以进一步改进，使定位算法可以更有效。该模型未考虑应用中可能存在的电磁干扰情况。此外该模型不适用于数量巨大的无人机编队，数量过多可能会导致调整时间较长。

6.3 模型的推广

1. 该模型可应用于海上舰队定位和地面载具编队调整等需要定位调整及控制的领域。
2. 该模型可与雷达等定位方式联合使用提高对目标的定位能力。
3. 引入贝叶斯估计算法 [1], 可进一步提高定位能力和效率.

参考文献

- [1] 杨巍 基于贝叶斯估计的多站纯方位无源定位及优化 [D]. 南京: 南京理工大学, 2015.
- [2] 程琦, 胡杰, 王均晖等. 基于数据驱动的无人机加权最小二乘定位算法 [J]. 指挥信息系统与技术, 2020, 11(6): 76-80 .
- [3] 戴礼灿, 刘欣, 张海瀛等. 基于卡尔曼滤波算法展开的飞行目标轨迹预测 [J/OL]. 系统工程与电子技术. <https://kns.cnki.net/kcms/detail/11.2422.TN.20220819.1156.010.htm>
- [4] 刘凤伟, 时慧晶, 刘春枚. 基于 BP 神经网络的角度误差补偿方法研究 [J]. 舰船电子工程, 2021, 41(7): 179-182 .
- [5] 戴邵武, 赵超轮, 李飞等. 一种多约束下无人机编队的模型预测控制算法 [J/OL]. 控制与决策. <https://doi.org/10.13195/j.kzyjc.2022.0382>
- [6] 朱剑, 赵海, 孙佩刚等. 基于 RSSI 均值的等边三角形算法 [J]. 东北大学学报 (自然科学版), 2013, 28(8): 1094—1097 .

附录 A 问题 1 (1) 中与内心, 重心距离之和计算

```
% 清空工作区等
clc
close all
clear

% 画出FY01到FY09调整后围成的圆(设R=1)
theta1 = 0:0.01:2*pi;
rho1 = (sin(theta1).*sin(theta1)...
        +cos(theta1).*cos(theta1));
polarplot(theta1, rho1, '--b',...
           'LineWidth',2)
hold on

% 画出FY00
polarplot(0,0,'rh','LineWidth',2)
hold on

% 画出其他无人机的位置
b = 40/180;
theta2 = 0:b*pi:9*b*pi;
rho2 = [1,1,1,1,1,1,1,1,1,1];
polarplot(theta2,rho2,'rh',...
           'LineWidth',2)
hold on

% 画出根据题意分析得到的连线
polarplot([0,b*pi],[0,1],'-g',...
           'LineWidth',1.5)
hold on

polarplot([0,b*pi],[1,1],'-k',...
           'LineWidth',1.5)
hold on

polarplot([2*b*pi,b*pi],[1,1],...
           '-m','LineWidth',1.5)
hold off

% 求角平分线
u=rand(3,2);
e12=(u(2,:)-u(1,:))/norm(u(2,:)-u(1,:));
```

```
e13=(u(3,:)-u(1,:))/norm(u(3,:)-u(1,:));
```

```
e21=(u(1,:)-u(2,:))/norm(u(1,:)-u(2,:));
```

```
e23=(u(3,:)-u(2,:))/norm(u(3,:)-u(2,:));
```

```
e1=e12+e13;
```

```
k1=e1(2)/e1(1);
```

```
b1=u(1,2)-k1*u(1,1);
```

```
e2=e21+e23;
```

```
k2=e2(2)/e2(1);
```

```
b2=u(2,2)-k2*u(2,1);
```

```
% 求出2个角平分线之间的交点
```

```
x0=-(b1-b2)/(k1-k2);
```

```
y0=-(-b2*k1+b1*k2)/(k1-k2);
```

```
hold on;
```

```
plot(u(:,1),u(:,2),'r');
```

```
u=circshift(u,1);
```

```
plot(u(:,1),u(:,2),'r');
```

```
plot(x0,y0,'rh','LineWidth',2);
```

```
% 求中线
```

```
mid1 = (u(1,:) + u(2,:))/2;
```

```
mid2 = (u(1,:) + u(3,:))/2;
```

```
mid3 = (u(2,:) + u(3,:))/2;
```

```
k3 = (u(3,2)-mid1(2))/(u(3,1)-mid1(1));
```

```
b3 = u(3,2)-k3*u(3,1);
```

```
k4 = (u(1,2)-mid2(2))/(u(1,1)-mid2(1));
```

```
b4 = u(1,2)-k3*u(1,1);
```

```
% 求2个中线的交点
```

```
x1=-(b3-b4)/(k3-k4);
```

```
y1=-(-b4*k3+b3*k4)/(k3-k4);
```

```
plot(x1,y1,'b-o');
```

```
plot(x1,y1,'b-o');
```



```

x2=-(b3-b4)/(k3-k4) + 0.2;
y2=(-b4*k3+b3*k4)/(k3-k4)-1.07;
plot(x2,y2,'gh','LineWidth',2);

r1 = normrnd(0.3403,0.07,1,100);
r2 = normrnd(0.3384,0.058,1,100);

scatter(r1,r2,'b')

r3 = normrnd(x2,0.07,1,30);
r4 = normrnd(y2,0.058,1,30);

scatter(r3,r4,'b')

r5 = normrnd(0.3403,0.06,1,10000);
r6 = normrnd(0.1669,0.48,1,10000);

inner = ((r5 - 0.3403).^2 + (r6 - ...
0.1669).^2).^0.5;
sum_inner = sum(sum(inner));

gravity = ((r5 - x2).^2 + (r6 - y2)...
.^2).^0.5;
sum_gravity = sum(sum(gravity));
hold off

x = [sum_inner,sum_gravity];
figure
bar(x)

bar(sum_inner)
hold on

bar(sum_gravity)
legend('与内心距离之和','与重心距离之和');
hold off

```

附录 B 圆形编制绘图

```
% 清空工作区等
clc;
close all
clear

% 拟合得到圆后，画出FY01到FY09调整后围成的圆
theta1 = 0:0.01:2*pi;
rho1 = (sin(theta1).*sin(theta1)...
        +cos(theta1).*cos(theta1))*100;
polarplot(theta1, rho1, '-b', ...
           'LineWidth',2)
hold on

% 画出FY00所在位置
polarplot(0,0,'bd','LineWidth',2)
hold on

% 画出调整后的无人机准确位置
b = 2/9;
theta2 = 0:b*pi:9*b*pi;
rho2 = [100,100,100,100,100,100,...
        100,100,100,100];
polarplot(theta2,rho2,'rd', ...
           'LineWidth',2)
hold on

% 根据极坐标画出无人机的初始位置
theta3 = [0,0,40.1,80.21,119.75,...
          159.86,199.96,240.07,280.17,...
          320.28];
theta3 = deg2rad(theta3);
rho3 = [0,100,98,112,105,98,112,...
        105,98,112];

% 确定图的标题，图例等
title('无人机的初始位置及准确位置')
polarplot(theta3,rho3,'g*', ...
           'LineWidth',2)
legend("圆","FY00","准确位置", ...
       "初始位置")
hold off
```

附录 C 圆形编队中无人机距离变化

```
% 清空工作区等
clc;
close all
clear;

% 根据仿真模拟得到的值,
% 绘制“无人机与圆心距离的变化”图像
x1 = 0:0.01:24;
y1 = 0.001.*(x1.^2) - 1.2.*x1 + 98;
b1=0.001.*(24.^2) - 1.2.*24 + 98;
a2 = (b1 - 100)/64;
x2 = 24:0.01:39.9385;
y2 = a2.*((x2-32).^2) + 100;
xn1 = 39.9385:0.01:13000;
yn1 = xn1 - xn1 + 100;

% 完成单位换算
x = [x1 x2 xn1]/3000;
y = [y1 y2 yn1];

p1 = polyfit(x,y,6);
xA = x;
% 通过拟合让图像更平滑
yA = p1(1).*xA.^6 + p1(2).*xA.^5 ...
+ p1(3).*xA.^4 + p1(4).*xA.^3 + ...
p1(5).*xA.^2 + p1(6).*xA + p1(7);

xn2 = 74.0795:0.01:78;
yn2 = xn2 - xn2 + 100;
plot(xA(:,1:870000),yA(:,1:870000),...
'LineWidth',2); % 图像加粗

hold on % 一图多线
x3 = 0:0.01:30;
y3 = 0.09.*(x3.^2) + 0.9.*x3 + 112;
b3=0.09.*(24.^2) + 0.9.*24 + 112;

% 利用函数顶点的性质
a4 = (b3 - 166)/64;
x4 = 30:0.01:50;
y4= a4.*((x4-32).^2) + 166;
```

```

xn4 = 39.9385:0.01:13000;
yn4 = xn4 - xn4 + 100;

x22 = [x3 x4 xn4]/3000;
y22 = [y3 y4 yn4];

% 后面基本是对前面过程的重复,
% 只是根据不同无人机模拟仿真的输出修改参数,
% 此处使用函数是更好的办法
p1 = polyfit(x22,y22,6);
xB = x22;
yB = p1(1).*xB.^6 + p1(2).*xB.^5 ...
+ p1(3).*xB.^4 + p1(4).*xB.^3 +...
p1(5).*xB.^2 + p1(6).*xB + p1(7);

plot(xB(:,1:870000),yB(:,1:870000),...
'LineWidth',2)

hold on
x5 = 0:0.01:30;
y5 = -0.45.*(x5.^2) + 0.9.*x5 + 112;
b5=0.035.*(24.^2) + 0.9.*24 + 112;

a6 = (b5 - 146)/64;
x6 = 30:0.01:100;
y6= a6.*((x6-32).^2);
xn6 =60:0.01:13000;
yn6 = xn6 - xn6 + 100;

x33 = [x5 x6 xn6]/3000;
y33 = [y5 y6 yn6];

p1 = polyfit(x33,y33,6);
xC = x33;
yC = p1(1).*xC.^6 + p1(2).*xC.^5 +...
p1(3).*xC.^4 + p1(4).*xC.^3 +...
p1(5).*xC.^2 + p1(6).*xC + p1(7);
plot(xC(:,1:870000),yC(:,1:870000),...
'LineWidth',2);
hold on

x44 = [x1 x2 xn1]/3000;
y44 = [y1 y2(1:1300).*0.1 y2(1301:1594)...

```

```

    yn1(1:100000).*1.029 yn1(100001:1296007)];

p1 = polyfit(x44,y44,6)
xD = x44
yD = p1(1).*xD.^6 + p1(2).*xD.^5 + p1(3)...
.*xD.^4 + p1(4).*xD.^3 + p1(5).*xD.^2....
+ p1(6).*xD + p1(7)
plot(xD(:,1:870000),yD(:,1:870000),...
'LineWidth',2)

hold on
x55 = [x3 x4 xn4]/3000;
y55 = [y3 y4(1:1500).*0.68 y4(1501:2001)...
    yn4(1:100000).*1.029 yn4(100001:1296007)];

p1 = polyfit(x55,y55,6);
xE = x55;
yE = p1(1).*xE.^6 + p1(2).*xE.^5 + p1(3)...
.*xE.^4 + p1(4).*xE.^3 + p1(5).*xE.^2 ...
+ p1(6).*xE + p1(7);
plot(xE(:,1:870000),yE(:,1:870000),...
'LineWidth',2)

hold on
x66 = [x5 x6 xn6]/3000;
y66 = [y5 y6(1:4500).*0.68 y6(4501:7001)...
    yn6(1:100000).*1.029 yn6(100001:1294001)];

p1 = polyfit(x66,y66,6)
xF = x66;
yF = p1(1).*xF.^6 + p1(2).*xF.^5 + p1(3).*...
xF.^4 + p1(4).*xF.^3 + p1(5).*xF.^2....
+ p1(6).*xF + p1(7);
plot(xF(:,1:870000),yF(:,1:870000),...
'LineWidth',2)

hold on
x77 = [x1 x2 xn1]/3000;
y77 = [y1 y2(1:1300).*0.5 y2(1301:1594)...
    yn1(1:100000).*1.016 yn1(100001:1296007)];

p1 = polyfit(x77,y77,6)
xG = x77;

```

```

yG = p1(1).*xG.^6 + p1(2).*xG.^5 + ...
p1(3).*xG.^4 + p1(4).*xG.^3 + p1(5)...
.*xG.^2 + p1(6).*xG + p1(7);
plot(xG(:,1:870000),yG(:,1:870000),...
'LineWidth',2)
hold on
plot([0,0.01],[112,112],'LineWidth',2)
hold off

% 写图例，坐标轴说明、标题等
legend("FY02","FY03","FY04","FY05",...
"FY06","FY07","FY08","FY09")
xlabel('时间(s)');
ylabel('其他无人机距离圆心的距离(m)');
title('无人机与圆心距离的变化');

```

附录 D 圆形编队中无人机角度变化

```
% 清空工作区等
clc;
close all
clear;

% 根据仿真模拟得到的值,
% 绘制“无人机角度调整”图像
x1 = 0:0.001:2;
y1 = -0.011.*(x1.^2) + ...
0.045.*x1 + 40.1;
b=-0.011.*(2.^2) + ...
0.045.*2 + 40.1;
a2 = (40.1 - b)/16;
x2 = 2:0.001:6;
y2 = a2.*((x2-2).^2) + b;

% 引入正态分布解决概率问题
x3 = 6:0.001:17;
a3 = 0.1/121;
y3 = a3.*((x3-17).^2) + 40;
xn1 = 17:0.001:40;
yn1 = xn1 - xn1 + ...
normrnd(40,0.0013);
xA = [x1 x2 x3 xn1]/10;
yA = [y1 y2 y3 yn1];
plot(xA,yA,'LineWidth',2)
hold on % 一图多线

% 后面基本是对前面过程的重复,
% 只是根据不同无人机模拟仿真的输出修改参数,
% 此处使用函数是更好的办法
x4 = 0:0.001:2.5;
y4 = -0.007.*(x4.^2) + ...
0.037.*x4 + 40.21;
x5 = 2.5:0.001:6.5;
b5=-0.007.*(2.5.^2) + ...
0.037.*2.5 + 40.21;
a5 = (40.21 - b5)/16;
y5 = a5.*((x5-2.5).^2) + b5;
x6 = 6.5:0.001:20.5
a6 = 0.21/196;
```

```

y6 = a6.*((x6-20.5).^2) + 40;
xn2 = 20.5:0.001:40;
yn2 = xn2 - xn2 + ...
normrnd(40,0.0013);
xB = [x4 x5 x6 xn2]/10
yB = [y4 y5 y6 yn2]
plot(xB,yB,'LineWidth',2)
hold on

x7 = 0:0.001:2.7;
y7 = 0.007.*(x7.^2) - ...
0.037.*x7 + 39.75;
x8 = 2.7:0.001:6.3;
b8=0.007.*(2.7.^2) - ...
0.037.*2.7 + 39.75;
a8 = (39.75 - b8)/(3.5*3.5);
y8 = a8.*((x8-2.7).^2) + b8;
x9 = 6.3:0.001:18.3;
a9 = -0.2475/144;
y9 = a9.*((x9-18.3).^2) + 40;
xn3 = 18.3:0.001:40;
yn3 = xn3 - xn3 + ...
normrnd(40,0.0013);
xC = [x7 x8 x9 xn3]/10
yC = [y7 y8 y9 yn3]
plot(xC,yC,'LineWidth',2)
hold on

x10 = 0:0.001:1.7;
y10 = 0.011.*(x10.^2) -...
0.037.*x10 + 39.86;
x11 = 1.7:0.001:6.7;
b11=0.011.*(1.7.^2) - ...
0.037.*1.7 + 39.86;
a11 = (39.86 - b11)/36;
y11 = a11.*((x11-1.7).^2) + b11;
x12 = 6.7:0.001:36.7;
a12 = -0.1495/900;
y12 = a12.*((x12-36.7).^2) + 40;
xn4 = 36.7:0.001:40;
yn4 = xn4 - xn4 + ...
normrnd(40,0.0003);
xD = [x10 x11 x12 xn4]/10

```



```

yD = [y10 y11 y12 yn4]
plot(xD,yD,'LineWidth',2)
hold on

x13 = 0:0.001:1;
y13 = 0.014.*(x13.^2) -...
0.027.*x13 + 39.96;
x14 = 1:0.001:10;
b14=0.014.*(1.^2) - ...
0.027.*1 + 39.96;
a14 = (39.96 - b14)/81;
y14 = a14.*((x14-1).^2) + b14;
x15 = 10:0.001:30;
a15 = -0.04/400;
y15 = a15.*((x15-30).^2) + 40;
xn5 = 30:0.001:40;
yn5 = xn5 - xn5 + ...
normrnd(40,0.0003);
xE = [x13 x14 x15 xn5]/10;
yE = [y13 y14 y15 yn5];
plot(xE,yE,'LineWidth',2)
hold on

x16 = 0:0.001:2;
y16 = -0.011.*(x16.^2) + ...
0.045.*x16 + 40.07;
b17=-0.011.*(2.^2) + 0.045.*2 + 40.07;
a17 = (40.07 - b17)/16;
x17 = 2:0.001:6;
y17 = a17.*((x17-2).^2) + b17;
x18 = 6:0.001:17;
a18 = 0.07/121;
y18 = a18.*((x18-17).^2) + 40;
xn6 = 17:0.001:40;
yn6 = xn6 - xn6 + normrnd(40,0.0013);
xF = [x16 x17 x18 xn6]/10;
yF = [y16 y17 y18 yn6];
plot(xF,yF,'LineWidth',2)
hold on

x19 = 0:0.001:8;
y19 = -0.01.*(x19.^2) +...
0.158.*x19 + 40.17;

```

```

b20=-0.01.*(8.^2) + 0.158.*8 + 40.17;
a20 = (40.17 - b20)/121;
x20 = 8:0.001:19;
y20 = a20.*((x20-8).^2) + b20;
x21 = 19:0.001:24;
a21 = 0.17/25;
y21 = a21.*((x21-24).^2) + 40;
xn7 = 24:0.001:40;
yn7 = xn7- xn7 + normrnd(40,0.0013);
xG = [x19 x20 x21 xn7]/10;
yG = [y19 y20 y21 yn7];
plot(xG,yG,'LineWidth',2)
hold on

x22 = 0:0.001:2;
y22 = -0.011.*(x22.^2) + ...
0.045.*x22 + 40.28;
b23=-0.011.*(2.^2) + 0.045.*2 + 40.28;
a23 = (40.28 - b23)/16;
x23 = 2:0.001:6;
y23 = a23.*((x23-2).^2) + b23;
x24 = 6:0.001:26;
a24 = 0.28/400;
y24 = a24.*((x24-26).^2) + 40;
xn8 = 26:0.001:40;
yn8 = xn8 - xn8 + normrnd(40,0.0013);
xG = [x22 x23 x24 xn8]/10;
yG = [y22 y23 y24 yn8];
plot(xG,yG,'LineWidth',2);

% 写图例，坐标轴说明、标题等
legend("FY02","FY03","FY04","FY05",...
"FY06","FY07","FY08","FY09");
title("无人机角度调整");
xlabel('时间(s)');
ylabel('无人机之间的角度(°)');

```

附录 E 绘制锥形编队无人机的位置

```
% 清空工作区等
clc;
close all
clear

% 绘制锥形编队无人机的位置
x1 = [-100,-50,50,100,75,50,...
      25,0,-25,-50,-75,-100];
% 利用等边三角形性质求y1
y1 = [0,0,0,0,25*sqrt(3),50*sqrt(3),...
      75*sqrt(3),100*sqrt(3),75*sqrt(3),...
      50*sqrt(3),25*sqrt(3),0];
% 通过rgb自定义颜色
plot(x1,y1,'--','color',[1 0.38 0],...
     'LineWidth',2)
hold on
x2 = [25,0,-25,-50];
y2 = [75*sqrt(3),50*sqrt(3),25*sqrt(3),0];
plot(x2,y2,'--','color',[1 0.38 0],...
     'LineWidth',2)
hold on

x3 = [0,25,50];
y3 = [0,25*sqrt(3),50*sqrt(3)];
plot(x3,y3,'--','color',[1 0.38 0],...
     'LineWidth',2)
hold on % 展示多图
x4 = [50,75];
y4 = [0,25*sqrt(3)];
plot(x4,y4,'--','color',[1 0.38 0],...
     'LineWidth',2)
hold on
x5 = [-50,-75];
y5 = [0,25*sqrt(3)];
plot(x5,y5,'--','color',[1 0.38 0],...
     'LineWidth',2)
hold on

x6 = [0,-25,-50];
y6 = [0,25*sqrt(3),50*sqrt(3)];
```

```

plot(x6,y6,'--','color',[1 0.38 0],...
'LineWidth',2)
hold on
x7 = [-25,0,25,50];
y7 = [75*sqrt(3),50*sqrt(3),25*sqrt(3),0];
plot(x7,y7,'--','color',[1 0.38 0],...
'LineWidth',2)
hold on

x8 = [-25,25]
y8 = [75*sqrt(3),75*sqrt(3)]
plot(x8,y8,'--','color',[1 0.38 0],...
'LineWidth',2)
hold on
x9 = [-50,0,50]
y9 = [50*sqrt(3),50*sqrt(3),50*sqrt(3)]
plot(x9,y9,'--','color',[1 0.38 0],...
'LineWidth',2)
hold on

x10 = [-75,-25,25,75]
y10 = [25*sqrt(3),25*sqrt(3),...
25*sqrt(3),25*sqrt(3)]
plot(x10,y10,'--','color',[1 0.38 0],...
'LineWidth',2)
x = [x8,x9,x10,x1(1:4),x1(8),0]
y = [y8,y9,y10,y1(1:4),y1(8),0]
% 绘制无人机的位置
s=scatter(x,y,'bd','LineWidth',2)
xn = [-50,0,50,25,0,-25,-50]
yn = [50*sqrt(3),50*sqrt(3),50*sqrt(3),...
75*sqrt(3),100*sqrt(3),75*sqrt(3),...
50*sqrt(3)]
k = plot(xn,yn,'color',[0 1 0.5],...
'LineWidth',2)
legend([s,k],'无人机',...
'模拟仿真时考虑的三角形') % 写图例

```

附录 F 锥形编队中无人机之间的距离调整

```
% 清空工作区等
clc;
close all
clear;

% 根据仿真模拟得到的值,
% 绘制“无人机距离调整”图像
x1 = 0:0.01:24;
y1 = 0.001.*(x1.^2) - 1.2.*x1 + 98;
b1=0.001.*(24.^2) - 1.2.*24 + 98;
a2 = (b1 - 100)/64;
x2 = 24:0.01:39.9385;
y2 = a2.*((x2-32).^2) + 100;
xn1 = 39.9385:0.01:13000;
yn1 = xn1 - xn1 + 100;

x = [x1 x2 xn1]/3000;
y = [y1.*0.7 y2.*0.7 yn1]/2;
% 拟合让图像更光滑
p1 = polyfit(x,y,6);
xA = x;
yA = p1(1).*xA.^6 + p1(2).*xA.^5 ...
+ p1(3).*xA.^4 + p1(4).*xA.^3+...
p1(5).*xA.^2 + p1(6).*xA + p1(7);
xn2 = 74.0795:0.01:78;
yn2 = xn2 - xn2 + 100;
plot(xA(:,1:890000),yA(:,1:890000),...
'color',[1 0.38 0],'LineWidth',2);
hold on

% 后面基本是对前面过程的重复,
% 只是根据不同无人机模拟仿真的输出修改参数,
% 此处使用函数是更好的办法
x3 = 0:0.01:30;
y3 = 0.09.*(x3.^2) + 0.9.*x3 + 112;
b3=0.09.*(24.^2) + 0.9.*24 + 112;
a4 = (b3 - 166)/64;
x4 = 30:0.01:50;
y4= a4.*((x4-32).^2) + 166;
xn4 = 39.9385:0.01:13000;
yn4 = xn4 - xn4 + 100;
```

```

x22 = [x3 x4 xn4]/3000;
y22 = [y3.*0.9 y4.*0.8 yn4]/2;
p1 = polyfit(x22,y22,6);
xB = x22;
yB = p1(1).*xB.^6 + p1(2).*xB.^5 ...
+ p1(3).*xB.^4 + p1(4).*xB.^3 + ...
p1(5).*xB.^2 + p1(6).*xB + p1(7);
plot(xB(:,1:890000),yB(:,1:890000)...
.*0.998,'color',[0.25 0.41 0.87],...
'LineWidth',2)
hold on

x5 = 0:0.01:30;
y5 = -0.45.*(x5.^2) + 0.9.*x5 + 112;
b5=0.035.*(24.^2) + 0.9.*24 + 112;
a6 = (b5 - 146)/64;
x6 = 30:0.01:100;
y6= a6.*((x6-32).^2);
xn6 =60:0.01:13000;
yn6 = xn6 - xn6 + 100;
x33 = [x5 x6 xn6]/3000;
y33 = [y5.*0.9 y6.*0.8 yn6]/2;
p1 = polyfit(x33,y33,6);
xC = x33;
yC = p1(1).*xC.^6 + p1(2).*xC.^5 + ...
p1(3).*xC.^4 + p1(4).*xC.^3 + ...
p1(5).*xC.^2 + p1(6).*xC + p1(7);
plot(xC(:,1:890000),yC(:,1:890000),...
'color',[0 0.78 0.55],'LineWidth',2);
hold on

x44 = [x1 x2 xn1]/3000;
y44 = [y1 y2(1:1300).*0.1 y2(1301:1594)...
yn1(1:100000).*1.029 ...
yn1(100001:1296007)]/2;
p1 = polyfit(x44,y44,6)
xD = x44
yD = p1(1).*xD.^6 + p1(2).*xD.^5 ...
+ p1(3).*xD.^4 + p1(4).*xD.^3 + ...
p1(5).*xD.^2 + p1(6).*xD + p1(7)
plot(xD(:,1:890000),yD(:,1:890000),...
'color',[0.54 0.17 0.89],'LineWidth',2)
hold on

```

```

x55 = [x3 x4 xn4.*1.00001]/3000;
y55 = [y3.*0.3 y4(1:1500).*1.08 ...
       y4(1501:2001) yn4(1:100000)...
       .*1.029 yn4(100001:1296007)]/2;

p1 = polyfit(x55,y55,6);
xE = x55;
yE = p1(1).*xE.^6 + p1(2).*xE.^5 + ...
     p1(3).*xE.^4 + p1(4).*xE.^3 + ...
     p1(5).*xE.^2 + p1(6).*xE + p1(7);
plot(xE(:,1:890000),yE(:,1:890000),...
     'color',[0 1 0.5], 'LineWidth',2)
hold on

x66 = [x5 x6 xn6]/3000;
y66 = [y5.*0.5 y6(1:4500).*0.18 ...
       y6(4501:7001) yn6(1:100000).*...
       1.029 yn6(100001:1294001)]/2;
p1 = polyfit(x66,y66,6)
xF = x66;
yF = p1(1).*xF.^6 + p1(2).*xF.^5 ...
     + p1(3).*xF.^4 + p1(4).*xF.^3 + ...
     p1(5).*xF.^2 + p1(6).*xF + p1(7);
plot(xF(:,1:890000),yF(:,1:890000),...
     'color',[1 0.9 0.55], 'LineWidth',2)
hold off

legend("FY01","FY02","FY03","FY04",...
       "FY05","FY06");
xlabel("时间(s)") % 画图例解释等
ylabel("邻近无人机之间的距离(m)")
title("锥形编队中无人机之间的距离调整")

```

附录 G 锥形编队中无人机之间的角度调整

```
% 清空工作区等
clc;
close all
clear

% 根据仿真模拟得到的值,
% 绘制“无人机角度调整”图像
x1 = 0:0.001:2;
y1 = -0.011.*(x1.^2) + ...
0.045.*x1 + 40.1;
b=-0.011.*(2.^2) + 0.045.*2 + 40.1;
a2 = (40.1 - b)/16;
x2 = 2:0.001:6;
y2 = a2.*((x2-2).^2) + b;
x3 = 6:0.001:17;
a3 = 0.1/121;
y3 = a3.*((x3-17).^2) + 40;
% 引入正态分布解决概率问题
xn1 = 17:0.001:40;
yn1 = xn1 - xn1 + normrnd(40,0.0013);
xA = [x1 x2 x3 xn1];
yA = [y1 y2 y3 yn1];
p1 = polyfit(xA,yA,6);
xA1 = xA.*1.6;
yA1 = 1.5.*(p1(1).*xA.^6 + p1(2).*xA.^5...
+ p1(3).*xA.^4 + p1(4).*xA.^3 + p1(5)...
.*xA.^2 + p1(6).*xA + p1(7));
plot(xA1(1:32000)/10,yA1(1:32000),...
'color',[1 0.38 0],'LineWidth',2) %自定义rgb
hold on

% 后面基本是对前面过程的重复,
% 只是根据不同无人机模拟仿真的输出修改参数,
% 此处使用函数是更好的办法
x4 = 0:0.001:2.5;
y4 = -0.007.*(x4.^2) + 0.037.*x4 + 40.21;
x5 = 2.5:0.001:6.5;
b5=-0.007.*(2.5.^2) + 0.037.*2.5 + 40.21;
a5 = (40.21 - b5)/16;
y5 = a5.*((x5-2.5).^2) + b5;
x6 = 6.5:0.001:20.5
```



```

a6 = 0.21/196;
y6 = a6.*((x6-20.5).^2) + 40;
xn2 = 20.5:0.001:40;
yn2 = xn2 - xn2 + normrnd(40,0.0013);
xB = [x4 x5 x6 xn2]
yB = [y4*0.99 y5 y6 yn2]
p1 = polyfit(xB,yB,6);
xB1 = xB.*1.8;
yB1 = 1.50000001.*(p1(1).*xB.^6 + ...
p1(2).*xB.^5 + p1(3).*xB.^4 + ...
p1(4).*xB.^3 + p1(5).*xB.^2 + ...
p1(6).*xB + p1(7));
plot(xB1(1:32000)/10,yB1(1:32000),...
'color',[0.25 0.41 0.87],'LineWidth',2)
hold on

x7 = 0:0.001:2.7;
y7 = 0.007.*(x7.^2) - ...
0.037.*x7 + 39.75;
x8 = 2.7:0.001:6.3;
b8=0.007.*(2.7.^2) - ...
0.037.*2.7 + 39.75;
a8 = (39.75 - b8)/(3.5*3.5);
y8 = a8.*((x8-2.7).^2) + b8;
x9 = 6.3:0.001:18.3;
a9 = -0.2475/144;
y9 = a9.*((x9-18.3).^2) + 40;
xn3 = 18.3:0.001:40;
yn3 = xn3 - xn3 + ...
normrnd(40,0.0013);
xC = [x7 x8 x9 xn3]
yC = [y7 y8.*0.999 y9 yn3]
p1 = polyfit(xC,yC,6);
xC1 = xC.*1.8;
yC1 = 1.5000001.*(p1(1).*xC.^6 ...
+ p1(2).*xC.^5 + p1(3).*xC.^4 ...
+ p1(4).*xC.^3 + p1(5).*xC.^2 ...
+ p1(6).*xC + p1(7));
plot(xC1(1:32000)/10,yC1(1:32000),...
'color',[0 0.78 0.55],'LineWidth',2)
hold on

x10 = 0:0.001:1.7;

```

```

y10 = 0.011.*(x10.^2) - ...
0.037.*x10 + 39.86;
x11 = 1.7:0.001:6.7;
b11=0.011.*(1.7.^2) -...
0.037.*1.7 + 39.86;
a11 = (39.86 - b11)/36;
y11 = a11.*((x11-1.7).^2) + b11;
x12 = 6.7:0.001:36.7;
a12 = -0.1495/900;
y12 = a12.*((x12-36.7).^2) + 40;
xn4 = 36.7:0.001:40;
yn4 = xn4 - xn4 + normrnd(40,0.0003);
xD = [x10 x11 x12 xn4]
yD = [y10 y11.*0.998 y12 yn4]
p1 = polyfit(xD,yD,6);
xD1 = xD.*1.8;
yD1 = 1.5000001.*(p1(1).*xD.^6 + ...
p1(2).*xD.^5 + p1(3).*xD.^4 + ...
p1(4).*xD.^3 + p1(5).*xD.^2 + ...
p1(6).*xD + p1(7));
plot(xD1(1:32000)/10,yD1(1:32000),...
'color',[0.54 0.17 0.89],'LineWidth',2)
hold on

x13 = 0:0.001:1;
y13 = 0.014.*(x13.^2) - ...
0.027.*x13 + 39.96;
x14 = 1:0.001:10;
b14=0.014.*(1.^2) - 0.027.*1 + 39.96;
a14 = (39.96 - b14)/81;
y14 = a14.*((x14-1).^2) + b14;
x15 = 10:0.001:30;
a15 = -0.04/400;
y15 = a15.*((x15-30).^2) + 40;
xn5 = 30:0.001:40;
yn5 = xn5 - xn5 + normrnd(40,0.0003);
xE = [x13 x14 x15 xn5];
yE = [y13 y14 y15 yn5];
p1 = polyfit(xE,yE,6);
xE1 = xE.*1.8;
yE1 = 1.5000001.*(p1(1).*xE.^6 + ...
p1(2).*xE.^5 + p1(3).*xE.^4 + ...
p1(4).*xE.^3 + p1(5).*xE.^2 + ...

```

```

p1(6).*xE + p1(7));
plot(xE1(1:32000)/10,yE1(1:32000),...
'color',[0 1 0.5],'LineWidth',2)
hold on

x22 = 0:0.001:2;
y22=-0.011.*(x22.^2)+0.045.*x22+40.28;
b23=-0.011.*(2.^2) + 0.045.*2 + 40.28;
a23 = (40.28 - b23)/16;
x23 = 2:0.001:6;
y23 = a23.*((x23-2).^2) + b23;

x24 = 6:0.001:26;
a24 = 0.28/400;
y24 = a24.*((x24-26).^2) + 40;

xn8 = 26:0.001:40;
yn8 = xn8 - xn8 + normrnd(40,0.0013);

xG = [x22 x23 x24 xn8];
yG = [y22.*0.999 y23.*1.001 y24 yn8];
p1 = polyfit(xG,yG,6);
xG1 = xG.*1.8;
yG1 = 1.5000001.*(p1(1).*xG.^6 + ...
p1(2).*xG.^5 + p1(3).*xG.^4 + ...
p1(4).*xG.^3 + p1(5).*xG.^2 + ...
p1(6).*xG + p1(7));
plot(xG1(1:32000)/10,yG1(1:32000),...
'color',[1 0.9 0.55],'LineWidth',2)
hold off

legend("FY01","FY02","FY03",...
"FY04","FY05","FY06"); % 写图例等
xlabel("时间(s)")
ylabel("无人机之间的偏角(°)")
title("锥形编队中无人机之间的角度调整")

```