

# 大作业 1：飞行器的多基雷达定位

## 摘要

现实生活中，使用雷达对飞行物进行定位的应用十分广泛，如何快速准确地得到目标飞行物的位置坐标并判断系统测量的精度，以及如何根据实测数据反向判断各个雷达的可靠性，都是十分重要的问题。本文基于超定方程的最小二乘解、MATLAB 模拟仿真、数据清洗等方法对飞行物的多基雷达定位问题进行研究，对上述问题提出了一些分析与见解。

对于问题一，首先对所需雷达个数进行分类讨论，由几何知识可定性分析，在雷达都在飞行物同一侧和雷达与飞行物高度关系不确定的情况下，分别至少需要 3 个和 4 个雷达。根据距离公式可知，多雷达测距求解飞行物定位问题可以转化为求超定方程组的解，因此可直接使用最小二乘法求得最优解。至于确定该算法精度，则可用 MATLAB 随机生成雷达坐标以及飞行物坐标位置，然后引入服从正态分布的误差，以超定方程组的最小二乘解与标准飞行物坐标之间的差作为对象，进行  $n$  次独立重复实验，获得其均值与标准差，并作为算法精度表示。

对于问题二，我们对雷达进行以基数为 3 的组合，取得所有可以获得唯一解的方程组集合，并取得所有唯一解的集合。我们对唯一解集合进行  $3\sigma$  法则处理，认定距离均值  $3\sigma$  范围外的解为异常解，找到构成此解的方程组与 3 个雷达，统计各个雷达出现的次数，当雷达出现次数高于某一阈值时，便认为这些雷达数据不可靠。

综上，我们提出了一种快速计算飞行物位置坐标的算法，并能够通过雷达所测数据反向识别问题雷达。最后我们对以上分析进行整合总结，对飞行器多基雷达定位问题提出了相关建议。

**关键字：** 多基雷达定位   飞行物定位   超定方程组   最小二乘法    $3\sigma$  法则

# 一、问题重述

## 1.1 问题背景与意义

随着航空技术的不断发展，不仅在军用方面，民用飞行器也越来越普遍。在此背景下，对于飞行器的空中管制愈发重要，而所面临的最基础的问题即为如何快速确定飞行物的精确位置。目前，国际四大卫星导航系统均采用三球交会的几何原理来实现定位，因此，研究不同条件下此方法的具体定位算法以及其精确度具有十分重要的价值。

## 1.2 问题相关信息

已知 3 个或 3 个以上雷达坐标  $R_i(x_i, y_i, z_i)$  以及各个雷达与飞行物距离  $r_i$ ，以非常快的速度求解飞行物坐标  $S(x, y, z)$ ，以便跟踪识别。

## 1.3 需要解决的问题

问题 (1): 对实际情况进行分类讨论，确定各个情况所需要的雷达个数

问题 (2): 在雷达个数确定的情况下，如何快速确定飞行物的空间坐标

问题 (3): 确定雷达的测量精度对快速算法的影响

问题 (4): 如果可以，设计反相辨识算法，辨识雷达数据的可靠性，并对快速算法进行优化，如果不能，便阐述原因

# 二、问题的分析

对于问题 (1) 与问题 (2)，飞行物在以雷达坐标  $R_i(x_i, y_i, z_i)$  为球心，雷达到飞行物的距离  $r_i$  为半径的球上，则我们只要知道对于不同情况，最少需要几个球相交可以确定唯一一点，即为各个情况所需雷达个数。同时，确定的唯一交点坐标即为飞行物坐标。

对于问题 (3) 与问题 (4)，我们认为雷达的定位误差与距离误差由许多相互独立的随机因素导致，所以根据中心极限定理，我们可以假设雷达的坐标误差以及雷达的测距误差在  $x, y, z$  方向上都服从独立的标准正态分布  $N(0, \sigma_r)$  和  $N(0, \sigma_1)$ 。然后确定雷达坐标误差与距离误差与飞行物坐标误差的关系，从而得到最终定位精度的表示方法，以及两种误差对精度的影响方式。

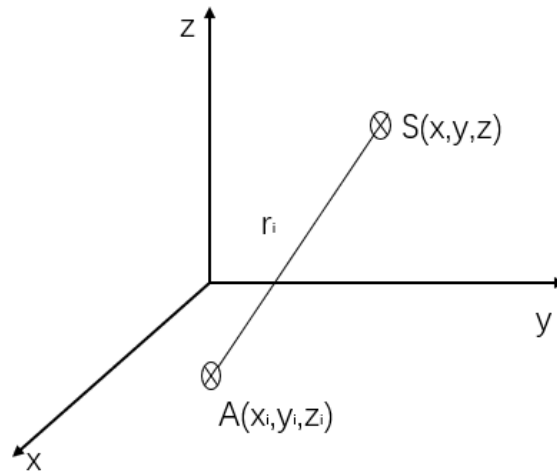


图 1 单个雷达定位示意图

### 三、基本假设与符号说明

#### 3.1 基本假设

- 假设各个雷达在同一瞬时获得测量结果，忽略时间差的影响
- 假设雷达与飞行物都为质点
- 假设误差服从正态分布

#### 3.2 符号说明

符号	意义
$R_i(x_i, y_i, z_i)$	第 $i$ 个雷达的位置坐标
$r_i$	第 $i$ 个雷达到飞行物的距离
$S(x, y, z)$	飞行物的真实空间坐标
$X$	最小二乘解得到的飞行物空间坐标
$Rd$	含有误差的雷达坐标
$rd$	含有误差的测量距离

### 四、问题一模型的建立与求解

任务：设计一个快速算法来确定这个飞行物的空间坐标，并确定算法精度。

## 4.1 问题一模型的建立

现已知雷达的坐标  $R_i(x_i, y_i, z_i)$  与雷达和飞行物之间的距离  $r_i$ ，因此飞行物的位置可以看作是在以雷达为球心，距离为半径的球面上。现有多个雷达同时测距，则飞行物应为所有球面的共同交点。其关系转化为数学公式可表示为：

$$\begin{cases} r_1^2 = (x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 \\ r_2^2 = (x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2 \\ \dots \\ r_i^2 = (x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 \end{cases} \quad (1)$$

因为飞行物的空间坐标为 3 个未知数，雷达与飞行物的距离关系为三元二次方程，所以至少需要 3 个方程才能确定飞行物的位置。问题转化为 3 个球面的公共点坐标。3 个球面如有公共点，则必有 2 个公共点，分别处于 3 个雷达所在平面的两侧，故为了能让 3 个球面确定唯一公共点，则必须已知 3 个雷达是否都低于飞行物或者都高于飞行物。否则，至少还需要一个球面来确定 2 个公共点的其中一个。对于此模型，我们分成两种情况讨论：

情况一：已知 3 个雷达是否都低于或都高于飞行物

情况二：雷达与飞行物的高度关系不确定

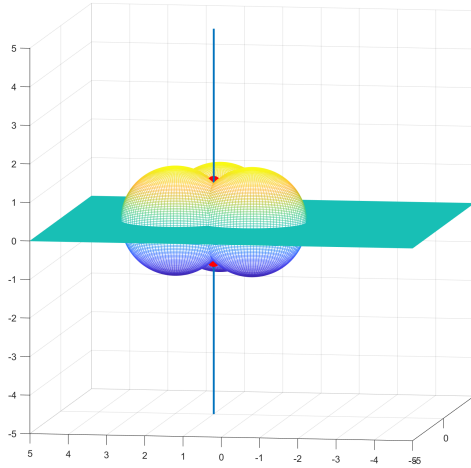


图 2 三球交汇示意图

### 4.1.1 已知雷达都低于飞行器高度或者都高于飞行器高度

#### (1) 雷达共线

以 3 个雷达所在直线为 x 轴建立直角坐标系，则  $y_1 = y_2 = y_3 = 0$ ， $z_1 = z_2 = z_3 = 0$ ，此时方程组变为：

$$\begin{cases} r_1^2 = (x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 \\ r_2^2 = (x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2 \\ r_3^2 = (x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2 \end{cases} \quad (2)$$

由式 2 可知，在此情况下无法求解出飞行物的  $y$  坐标与  $z$  坐标。

## (2) 雷达不共线

以三个雷达所在平面为  $xoy$  平面，建立直角坐标系，则  $z_1 = z_2 = z_3 = 0$ ，此时方程组变为：

$$\begin{cases} r_1^2 = (x - x_1)^2 + (y - y_1)^2 + z^2 \\ r_2^2 = (x - x_2)^2 + (y - y_2)^2 + z^2 \\ r_3^2 = (x - x_3)^2 + (y - y_3)^2 + z^2 \end{cases} \quad (3)$$

此时，由式 3 可确定两个解  $(x, y, z)$ ， $(x, y, -z)$ ，且已知雷达都低于飞行器高度或者都高于飞行器高度，若都高于，则唯一解为  $(x, y, -z)$ ，若都低于，则唯一解为  $(x, y, z)$ 。

### 4.1.2 雷达与飞行器高度关系不确定

在这种情况下，3 个雷达所得到的会有 2 个目标点，所以至少需要 4 个雷达才能确定唯一公共点。

#### (1) 雷达共面且至少有 3 个雷达不共线

以 4 个雷达所在平面为  $xoy$  平面建立直角坐标系，则  $z_1 = z_2 = z_3 = z_4 = 0$ ，此时方程组变为：

$$\begin{cases} r_1^2 = (x - x_1)^2 + (y - y_1)^2 + z^2 \\ r_2^2 = (x - x_2)^2 + (y - y_2)^2 + z^2 \\ r_3^2 = (x - x_3)^2 + (y - y_3)^2 + z^2 \\ r_4^2 = (x - x_4)^2 + (y - y_4)^2 + z^2 \end{cases} \quad (4)$$

可利用任意 3 个不共线雷达的方程组，即式 4，得到 2 个公共点坐标，对称于此三雷达所在平面， $(x, y, z)$ ， $(x, y, -z)$ 。由于这 4 个球面必有至少一个公共点。假设  $(x, y, z)$  在第四个球面上，由于都对称于  $xoy$  平面，则  $(x, y, -z)$  也在球面上。故在 3 个不共线雷达的基础上再加  $n$  个与之共面的雷达，也依然无法得出唯一解。

#### (2) 雷达不共面

以 4 个雷达其中 3 个雷达所在平面为  $xoy$  平面建立直角坐标系，则  $z_1 = z_2 = z_3 = 0$ ，此时方程组变为：

$$\begin{cases} r_1^2 = (x - x_1)^2 + (y - y_1)^2 + z^2 \\ r_2^2 = (x - x_2)^2 + (y - y_2)^2 + z^2 \\ r_3^2 = (x - x_3)^2 + (y - y_3)^2 + z^2 \\ r_4^2 = (x - x_4)^2 + (y - y_4)^2 + (z - z_4)^2 \end{cases} \quad (5)$$

根据式 5 前三个方程得到 3 个不共线雷达方程组，即式 3，得到 2 个公共点坐标， $(x, y, z)$ ， $(x, y, -z)$ 。由于此 4 个球面有至少一个公共点，所以其中至少有一点在第四个球面上，将 2 个点坐标都带入第四个方程得：

$$\begin{cases} r_4^2 = (x - x_4)^2 + (y - y_4)^2 + (z - z_4)^2 \\ r_4^2 = (x - x_4)^2 + (y - y_4)^2 + (-z - z_4)^2 \end{cases} \quad (6)$$

两个方程在  $z$  不等于 0 的情况下，只能有一个成立，所以此时，4 个方程能够得到唯一解。

## 4.2 问题一模型的求解

由于通常情况下雷达高度都低于飞行物，且雷达几乎不会在同一直线上，所以在这里我们只考虑  $n$  个低于飞行物且其中至少有 3 个不共线的雷达的情况，即最少 3 个雷达就可得到一个飞行物空间坐标数据，且数据集至少含有一个元素。因为方程组中方程数量多于未知数数量，可采用超定方程组的最小二乘法来求解：

### 4.2.1 飞行物坐标的求解

通过  $n$  个雷达，我们可以得到  $n$  个模型方程，即三元方程组式 1，将方程组中第二个方程以后的每个方程与第一个方程相减，所得如下：

$$\begin{cases} x_2^2 - x_1^2 + y_2^2 - y_1^2 + z_2^2 - z_1^2 + r_1^2 - r_2^2 = 2(x_2 - x_1)x + 2(y_2 - y_1)y + 2(z_2 - z_1) \\ x_3^2 - x_1^2 + y_3^2 - y_1^2 + z_3^2 - z_1^2 + r_1^2 - r_3^2 = 2(x_3 - x_1)x + 2(y_3 - y_1)y + 2(z_3 - z_1) \\ \dots \\ x_n^2 - x_1^2 + y_n^2 - y_1^2 + z_n^2 - z_1^2 + r_1^2 - r_n^2 = 2(x_n - x_1)x + 2(y_n - y_1)y + 2(z_n - z_1) \end{cases} \quad (7)$$

化为矩阵形式为：

$$\begin{bmatrix} 2(x_2 - x_1) & 2(y_2 - y_1) & 2(z_2 - z_1) \\ 2(x_3 - x_1) & 2(y_3 - y_1) & 2(z_3 - z_1) \\ \dots & & \\ 2(x_n - x_1) & 2(y_n - y_1) & 2(z_n - z_1) \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x_2^2 - x_1^2 + y_2^2 - y_1^2 + z_2^2 - z_1^2 + r_1^2 - r_2^2 \\ x_3^2 - x_1^2 + y_3^2 - y_1^2 + z_3^2 - z_1^2 + r_1^2 - r_3^2 \\ \dots \\ x_n^2 - x_1^2 + y_n^2 - y_1^2 + z_n^2 - z_1^2 + r_1^2 - r_n^2 \end{bmatrix} \quad (8)$$

简写为  $AX = b$ ，其中  $X = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$ 。

求解超定方程组的最小二乘解可得其最优解：

$$X' = (A^T A)^{-1} A^T b \quad (9)$$

最终求得飞行物的位置坐标。

算法代码如下：

```
% 算法一超定方程最小二乘解
% input Rd 含有误差的 m 个坐标集合
% input rd 含有误差的 m 个距离集合
% output X 计算所得最小二乘解
function X = overdetermined_equation(Rd,rd)
m = size(Rd,1); % 求得 m 个雷达
A = 2.*(Rd(2:m,:) - repmat(Rd(1,:),m-1,1));
b = sum(Rd(2:m,:).^2 - repmat(Rd(1,:),m-1,1).^2,2) - rd(2:m,:).^2 +
    repmat(rd(1,:),m-1,1).^2;

M = A'*b;
N = A'*A;
X = N\M;
end
```

#### 4.2.2 模型精度的求解

根据所查资料，雷达一般高度均低于 5 千米，因此设定飞行物飞行高度为 5 千米以上。

- 随机生成飞行物的坐标  $S(x, y, z)$  和  $m$  个雷达坐标  $R_i(x_i, y_i, z_i)$ ，并计算出飞行物与各雷达间的距离  $r_i$ ，将此处生成的坐标和距离作为参考标准
- 对雷达坐标和距离分别引入误差得到  $Rd, rd$ ，将此作为实际测量值

- 建立超定方程组求解最小二乘解  $X$
- 如上三步进行  $n$  次获得  $n$  组实际飞行物坐标  $S$  和计算所得飞行物坐标  $X$
- 对  $n$  组  $S$  和  $X$  的值求误差均值  $miu$  和标准差  $sigma$ ，即可描述该算法的精度

代码实现如下：

```
m = 10;      %设定雷达个数 m
n = 10000;
dist = 5000;
Xn = zeros(10,3);
Sn = zeros(10,3);

% 确定所设定算法一的精度
for i = 1:n
    [S,Rd,rd] = set_value(m,dist);
    X = overdetermined_equation(Rd,rd);
    Xn(i,:) = X;
    Sn(i,:) = S;
end

miu = mean((Xn-Sn),1); % 误差的平均值
sigma = std((Xn-Sn),0,1); % 误差的方差

% 生产标准结果
function [S,Rd,rd] = set_value(m,dist)
R = dist*rand(m,3); %产生标准雷达坐标
S = dist*rand(1,3)+5000; %产生标准飞行器坐标
r = sqrt(abs(sum((R-repmat(S,m,1)).^2,2))); %产生标准距离

Rd = R + sqrt(1)*randn(m,3); %引入坐标误差
rd = r + randn(m,1); %引入距离误差
end
```

### 4.3 问题一模型的检验与修正

根据上述算法求得误差均值与标准差作为精度的描述依据，我们发现通过更改雷达数量和雷达之间的距离可以显著改变误差的标准差。具体变化如表 1 与表 2 所示：

根据查阅多雷达数据融合相关资料，我们发现更改雷达数量与雷达间距离所得精度变化结果确实符合实际。



雷达数量 m/个	5	10	20	30	40	100
$\mu$	0.614	0.061	0.161	0.062	0.021	0.047
$\sigma$	14.870	4.359	3.199	2.937	2.775	2.491

表 1 雷达数量对精度的影响

雷达间距离 dist/米	2000	3000	4000	5000	7000	10000
$\mu$	0.077	0.107	0.166	0.052	0.055	0.047
$\sigma$	10.411	7.234	5.480	4.524	3.124	2.582

表 2 雷达之间的距离对精度的影响

根据数据模拟，我们发现当雷达个数从 5 个增加到 30 个时，误差标准差从 14 左右降低至了 2 左右，此后再增加雷达个数标准差变化则十分缓慢，同时，当雷达间距离从 2000 米到 10000 米增加时，误差标准差也有明显减小，因此采用 20 至 30 个雷达同时测量，且雷达间距离最好大于 5000 米较为科学。

## 五、问题二模型的建立与求解

任务：判断各个雷达不同的测距精度  $\epsilon_i$  对结果的影响，以及能否通过测得数据集反向判断雷达测距精度。

### 5.1 问题二模型的求解

此问题任然沿用问题一的模型，当各个雷达测距精度差别较大时，由于使用超定方程组的最小二乘法时无法对雷达数据进行数据清洗，导致计算所得的最优解与真实值相差较大。

因此，采用如下反向辨识算法来排除异常值：

- 获得超定方程组后，假设有  $n$  个方程，从中选取 3 个方程的组合，并对  $C_n^3$  个方程组进行求解，如无唯一解则舍弃该方程组，并最终建立唯一解集合
- 对唯一解集合采用  $3\sigma$  进行数据清洗，计算出唯一解集合的均值  $\mu$  和标准差  $\sigma$ ，并提取出在  $(\mu - \sigma, \mu + \sigma)$  范围外的解作为异常解，建立异常解集合

- 根据异常解反向查找到此解的方程组，并建立异常雷达集合，并根据雷达出现次数进行优先级排列
- 当雷达出现次数超过某一设定阈值，则认为此雷达数据可靠性低

核心代码实现如下：

```
function [Jnum,Jmis_num] = Judge_radar(mistake_re)
    num=unique(mistake_re(:))'; %汇总存在问题的雷达类型
    mis_num=histc(mistake_re(:),num); %计算各个雷达出错相应的次数

    miu_num = mean(mis_num); % 误差的平均值
    sigma_num = std(mis_num); % 误差的方差
    id = mis_num(:)>(miu_num+sigma_num);

    Jnum = num(id');
    Jmis_num = mis_num(id)';
end
```

## 5.2 问题二模型的检验与修正

根据上述算法，当某些雷达不准确度超过一定阈值时，我们可成功将其剔除。

引入距离误差之后，设定两个测量精度较大的雷达：

```
rd = r + randn(m,1); %引入距离误差
rd(2) = rd(2) + 3000; %设定存在问题的雷达
rd(10) = rd(10) + 3000; %设定存在问题的雷达
```

最初我们发现  $\mu \pm 3\sigma$  并不能有效筛选问题雷达，因为其范围较大，几乎涵盖了所有的唯一解。因此我们开始减小该范围，直到当范围变成  $\mu \pm \sigma$  时，算法才能够实现雷达的筛选。

之后我们调节问题雷达的误差，发现只有当误差大约大于 800 米时，问题雷达才能被较为准确地剔除。

很显然我们目前的算法依然存在问题，对于其修正方向，当获得唯一解集合之后，我们可以在空间中建立散点图，之后根据密度聚类分析 DBSCAN 算法对集合进行遍历，判断以每一个点为球心，半径为  $r$  的范围之内散点的个数，当该范围内的散点个数超过一定阈值时，便可将此处的球判定为一个聚类簇，具体如图 3 所示。最终当某一点不属于任何一个聚类簇时，便可认为其为异常值（图 3 中的 Noise point），由此可反向推导至其对应雷达。

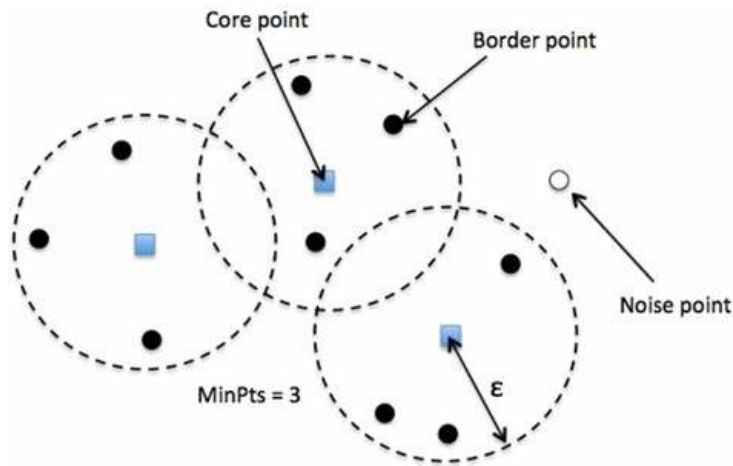


图3 DBSCAN 算法示意图

## 六、模型的评价

### 6.1 优点

- 采用物理三球交汇模型解释了雷达与飞行物之间的关系，更加形象直观、通俗易懂
- 采用超定方程组的最小二乘法求的最优解，算法速度快
- 评估测量精度时，采用了误差平均值与标准差两个衡量因素，使结果更加科学合理
- 采用  $3\sigma$  法则排除了不符合条件的异常值

### 6.2 缺点

- 问题一的检验过程中，设定的雷达之间的距离以及飞行高度不是很具有普适性
- 问题二中只有当雷达误差较大时才能被检测出来，若存在较小的误差还是无法分辨
- 问题二中算法使用了较多层循环，时间复杂度较高

## 参考文献

- [1] 杨晶, 万海宇, 何峰. 多基雷达对飞行目标的探测及攻击模型
- [2] 基于多基雷达目标定位的数学模型

## 附录 A 模拟真实的雷达数据

```
function [Rd,rd] = set_value(m)
    R = 5000*rand(m,3); %产生标准雷达坐标
    S = 5000*rand(1,3)+5000; %产生标准飞行器坐标
    r = sqrt(abs(sum((R-repmat(S,m,1)).^2,2))); %产生标准距离

    Rd = R + sqrt(1)*randn(m,3); %引入坐标误差
    rd = r + randn(m,1); %引入距离误差
    rd(2) = rd(2) + 3000; %设定存在问题的雷达
    rd(10) = rd(10) + 3000; %设定存在问题的雷达
end
```

## 附录 B 超定方程组的最小二乘解

```
function X = overdetermined_equation(Rd,rd)
m = size(Rd,1);
A = 2.*(Rd(2:m,:) - repmat(Rd(1,:),m-1,1));
b = sum(Rd(2:m,:).^2 - repmat(Rd(1,:),m-1,1).^2,2) - rd(2:m,:).^2 +
    repmat(rd(1,:),m-1,1).^2;

M = A'*b;
N = A'*A;
X = N\M;
end
```

## 附录 C 问题一算法精度的求解

```
m = 10; %设定雷达个数 m
n = 10000;
dist = 5000;
Xn = zeros(10,3);
Sn = zeros(10,3);
```

```

% 确定所设定算法一的精度
for i = 1:n
    [S,Rd,rd] = set_value(m,dist);
    X = overdetermined_equation(Rd,rd);
    Xn(i,:) = X;
    Sn(i,:) = S;
end
miu = mean((Xn-Sn),1); % 误差的平均值
sigma = std((Xn-Sn),0,1); % 误差的方差

% 生产标准结果
function [S,Rd,rd] = set_value(m,dist)
R = dist*rand(m,3); %产生标准雷达坐标
S = dist*rand(1,3)+5000; %产生标准飞行器坐标
r = sqrt(abs(sum((R-repmat(S,m,1)).^2,2))); %产生标准距离

Rd = R + sqrt(1)*randn(m,3); %引入坐标误差
rd = r + randn(m,1); %引入距离误差
end

```

## 附录 D 对雷达组进行基数为 3 的组合得到唯一解集合

```

m = 10; %设定雷达个数
Xn = [];
mistake_re = [];

function [Xn,mistake_re] = equations_sol(m,Rd,rd,Xn,mistake_re,x,y,z)
    result = [NaN NaN NaN];
    for i = 1:m
        eq1 = sum((Rd(i,:) - [x y z]).^2) - rd(i).^2 == 0;
        for j = i+1:m
            eq2 = sum((Rd(j,:) - [x y z]).^2) - rd(j).^2 == 0;
            for k = j+1:m
                eq3 = sum((Rd(k,:) - [x y z]).^2) - rd(k).^2 == 0;
                sol = solve(eq1,eq2,eq3,x,y,z);
                for g = 1:2

```

```

        a = vpa(sol.x(g,1));b = vpa(sol.y(g,1)); c = vpa(sol.z(g,1));
        if (a>0 && b >0 && c>0 && isreal(a))
            result = [double(vpa(sol.x(g,1))) double(vpa(sol.y(g,1)))
                double(vpa(sol.z(g,1)))];
            Xn = [Xn;result];
        elseif g == 2
            mistake_re = [mistake_re,[i j k]];
        end
    end
end
end
end
end
end

```

## 附录 E 对唯一解集合进行 $3\sigma$ 法则处理

```

function Xn = sigma3_method(Xn)
    miu1 = mean((Xn),1); % 误差的平均值
    sigma1 = std((Xn),0,1); % 误差的方差

    for i = 1:3
        id = (Xn(:,i)<(miu1(1,i)-sigma1(1,i))) |
            (Xn(:,i)>(miu1(1,i)+sigma1(1,i))); % 排除异常
            值
        Xn(id,:) = [];
    end

    miu2 = mean((Xn),1); % 误差的平均值
    sigma2 = std((Xn),0,1); % 误差的方差
end

```

## 附录 F 由异常解集合得到异常雷达集合

```

function [Jnum,Jmis_num] = Judge_radar(mistake_re)
    num=unique(mistake_re(:))'; %汇总存在问题的雷达类型

```

```
mis_num=histc(mistake_re(:),num); %计算各个雷达出错相应的次数

miu_num = mean(mis_num); % 误差的平均值
sigma_num = std(mis_num); % 误差的方差
id = mis_num(:)>(miu_num+sigma_num);

Jnum = num(id');
Jmis_num = mis_num(id)';
end
```