

参赛密码 _____
(由组委会填写)

“华为杯”第十三届全国研究生
数学建模竞赛

学 校	东南大学
参赛队号	10286199
队员姓名	1. 徐晓晨
	2. 王维舟
	3. 周欣

参赛密码 _____

(由组委会填写)



“华为杯”第十三届全国研究生 数学建模竞赛

题 目 基于无线通信基站的室内三维定位问题

摘 要：

在基于无线通信基站的室内三维定位问题中，一般是通过无线信号的到达时间（TOA）来计算手持终端到基站之间的距离。目前室内定位的精度不高，造成这种情况的主要原因是存在信号的非视距（NLOS）传播。NLOS 传播是指信号在从终端向基站传播的过程中会产生反射、折射甚至被遮挡等情况，这会对 TOA 的测量值产生较大的正误差。其次，如果终端与基站不在同一个“时间坐标系”中，即终端与基站的时钟不同步，则所测得的 TOA 数据就会有正误差和负误差。除此之外，干扰信号、噪声信号以及无线电波的多径传播都会对距离测量值产生影响。本文在利用 TOA 测量终端到基站的距离的基础上，基于最小二乘法进行改进，提出了基于非线性规划方法的加权模型。该模型利用终端与区域内多个基站之间的几何距离关系，构造距离误差的目标函数，并选取合理的距离测量值作为非线性约束函数和约束条件，从而提高定位的精度。在此基础上，采用基于三边测量的节点选择算法，选取尽可能少的基站数量实现近似最优定位，并从理论和实验数据两方面论证了六个基站实现定位的可靠性。在终端移动的场景下，本文采用改进的最小二乘法对终端的移动轨迹进行拟合，取得了较好的效果。本文还通过统计区域内可定位的终端数量，分析并得出了连接度数与定位精度之间的近似函数关系。

问题一：利用手持终端到各个基站的 TOA 测量数据和基站的三维坐标信息构建非线性规划模型。考虑到一些 TOA 测量数据是在 NLOS 环境中获得的，并且测

量值中存在许多误差因素。为减小这些误差对模型及精度的影响,本文基于最小二乘法进行改进,提出了对模型中的不同变量赋予不同权值的方法。通过将权值融合到目标函数中,来模拟各种误差对 TOA 测量数据的影响,从而大大提高了模型对三维空间中终端位置的定位精度。最后,将模型应用于题目中所给出的测试用例,将得到的结果与测试用例中的正确结果进行对比和误差分析,调整模型中的各项参数,以进一步提高模型的定位精度。

问题二:在问题一的基础上,使用尽可能少的基站数目,完成对移动终端的定位。通过基于三边测量的节点选择算法,确定在三维空间中定位一个终端至少需要四个基站。结合室内定位 NLOS 传播的场景,在对称分布的四个基站的基础上,选择两个与终端最接近 LOS 传播的基站作为约束条件,提高了移动终端定位的准确性。实验数据对比表明,基站数量增加到六个时可以很大程度的降低误差,实现近似最优的三维定位。

问题三:通过对于移动过程中的终端采集到的 TOA 数据,计算出各个时刻该终端的位置坐标。根据各时刻位置坐标的分布,用最小二乘法拟合出终端的运动轨迹。由于包含误差较大样本点在内所有样本点都参与运算,所以会对曲线拟合的最后结果产生偏差。针对这种情况,采用随机理论的思想,先随机选取 24 个点拟合曲线,然后计算与此曲线匹配的所有样本点个数。重复此过程一定次数,采用投票机制,匹配样本点多的椭圆即为最优曲线,构造了一种快速准确剔除误差较大样本点的改进曲线拟合算法。

问题四:在实际情况中,每个基站的通信范围都是有限的。假设基站的通信半径为 200 米,利用问题一中的模型对各终端进行定位后,依次计算其到各基站的距离,只有当距离在 200 米范围之内才认为此测量数据是有效的。同时利用问题二的结论,可以判断出能够被基站定位的所有终端数目。根据题目对“连接度数” λ 的定义,依次计算出五个场景中的 λ 值。在判断 λ 与定位精度之间的关系时,将“ λ 与定位精度之间的关系”转化为“ λ 与能够精确定位的终端数”之间的关系,因为一个场景中能够精确定位的终端数越多,则说明定位精度越高。最后通过曲线拟合的方法,得出两者的近似函数关系。

关键词: 室内定位; 非线性规划; 节点选择; 最小二乘法; 曲线拟合

目录

1 问题重述	1
1.1 问题背景	1
1.2 问题分析	1
2 模型假设	2
3 符号说明	2
4 问题一模型建立与求解	2
4.1 问题描述及分析	2
4.2 模型方法选择	3
4.3 模型建立	4
4.4 模型评价	6
4.5 对测量方式的改进	6
5. 问题二模型建立与分析	6
5.1 问题描述与分析	6
5.2 模型建立	7
5.3 数据分析	11
6 问题三模型建立与求解	13
6.1 问题描述及分析	13
6.2 模型建立	13
6.3 实验数据分析	17
6.4 模型评价	19
7 问题四模型建立与求解	20
7.1 问题描述及分析	20
7.2 模型建立	20
7.3 实验数据分析	20
8. 总结	24
参考文献	25
附录	25

1 问题重述

1.1 问题背景

随着智能无线终端，特别是智能手机的广泛普及，定位功能已经越来越多地应用于人们的日常生活中。目前，基于 GPS 的室外定位技术已经比较成熟，民用级别的 GPS 定位误差仅为 10 米左右。然而，GPS 信号对环境有一定要求，若想要获得较强的 GPS 信号，需要在比较空旷的室外环境中，因此室内定位技术很难利用 GPS 来实现。GPS 无法精确定位的区域一般是高楼林立的城区、建筑物内部、地下停车场等，而这些却是需要定位服务最多的区域。另外，随着最近几年移动通信的快速发展、移动基站的迅速普及，也使世界从网络时代进入无线互联时代，提供基于地理位置信息的服务已经成为最具市场前景和发展潜力的业务之一。基于无线移动通信网络的定位是以获取用户手持终端的位置为目标。而达成这一目标的手段是通过记录从手持终端发出的无线电信号到基站的传播时间，将时间乘以无线电信号的传播速度，即得到基站与终端之间沿某条路径的距离。最后利用定位算法将距离信息转化成终端的坐标信息。

1.2 问题分析

手持终端的三维坐标可以由最少四个基站到终端的距离来确定。终端的位置就在以基站坐标为球心，基站到终端的距离为半径的四个球形的相交区域。基站到终端的距离为 $d = t_r * c$ ，其中 $c = 3 \times 10^8 m/s$ ， t_r 为信号在基站与终端之间的真实传播时间。无线电信号的传播分为视距传播 (LOS) 和非视距传播 (NLOS)，NLOS 中信号要经过多次反射、折射、衍射才能到达，这导致 TOA 测量的时间数据中会掺杂 NLOS 误差。无线电信号的到达时间测量也会受到基站与终端之间时钟不同步的影响。另外，测量噪声误差也会对 TOA 的测量数据有影响。因此，我们需要在 TOA 测量数据中求出信号在基站到终端的真实传播时间，以此来得出基站到终端的距离。

使用尽可能少的基站对移动终端实现定位，理论上三维空间中四个球可以确定一个点的位置。但是由于信号造成的测量误差，需要添加约束条件来提高定位的准确度。因此，需要考察尽可能少的基站的数量以及其位置的空间分布，通过数据对比来说明其定位的可靠性。

通过对于移动过程中的终端采集到的 TOA 数据，计算出各个时刻该终端的位置坐标。根据各时刻位置坐标的分布，拟合出终端的运动轨迹。在位置坐标数据中，存在误差较大的数据，在拟合之前，我们需要对数据进行前期处理用来提高拟合精度。

问题四分为两部分。首先需要根据终端与各基站之间的距离来判断该终端能否被定位，并通过算法找出所有可以被基站定位的终端。其次，根据题目中对“连接度数” λ 的定义，需要计算出各个场景中的 λ 的值，并通过一定的模型找出 λ 与定位精度之间的关系。

2 模型假设

- (1) 假设终端进行的是长时间、有序的运动，而不是在短时间内无序的往复运动。
- (2) 假设给定区域内终端到每个基站的 TOA 测量值都是有效的。
- (3) 假设每个场景中既无线信号的 LOS 传播，也有 NLOS 传播。
- (4) 假设信号强度对 TOA 测量值无影响。
- (5) 假设基站所接收的信号是终端发出的信号，而不是干扰或噪声信号，即不会因为干扰或噪声的影响而造成给定的 TOA 数据小于真实值。

3 符号说明

α	目标函数中 X,Y 的权值
β	目标函数中 Z 的权值
B1	距离终端最近的一组基站
B2	距与终端距离相差不大的一组基站
Error	误差距离均值
Index_inliers	记录样本点编号
Num_inliers	匹配点总个数
Max_inliers	匹配点个数的最大值
Paras	记录曲线拟合参数
γ	平均连接度数
λ	连接度数
a	终端到基站之间连接数至少为 6 个的终端数

4 问题一模型建立与求解

4.1 问题描述及分析

根据基站的三维坐标(X_i, Y_i, Z_i)以及手持终端到基站的 TOA 测量数据计算出终端的三维坐标。由于信号传输环境十分复杂，很多情况下信号无法直射到达，即信号的视距 (LOS) 路径被阻挡，此时，信号需要经过多次反射、折射或衍射等非视距 NLOS 路径才能到达。这导致给定的 TOA 测量数据包含一般的测量噪声的同时还可能掺杂着由于 NLOS 路径所引起的误差以及由于基站时钟与终端时钟不同步导致的时区误差。一般来讲，NLOS 误差很大，其量级在 10^2 米。时钟不同步问题引起的误差大约在 100 米左右。而测量噪声的误差远小于 NLOS 误差和时间不同步误差。因此本文假定 TOA 测量数据由 NLOS 误差、LOS 路径时间以及时钟不同步误差组成，以此来简化模型。

在上述条件设定下，基站三维坐标及传播时间(TOA)测量数据已知，需要根据这些基站与终端之间的 TOA，给出当前终端的位置。结合问题假设，所建立的模型需要能在任意场景下，自适应的选择与场景匹配的模型预算法，并根据测量数据迅速的对终端进行准确定位。模型应当重点解决以下问题：

(1) 由于 NLOS 误差是无线定位系统中的主要来源之一，因此要对 NLOS 误差进行有效抑制。

(2) 移动终端三维坐标中的 Z 轴位置的确定。

4.2 模型方法选择

目前，NLOS 误差的抑制主要有两种思路：①间接法，其思路是在位置解算的过程中采用加权处理以减轻 NLOS 误差对定位结果的影响，从而增强算法在 LOS 与 NLOS 混合环境下的实用性，这种思路不需要对测量值进行预处理，但是要求参与定位的基站数目必须大于 3 个。②直接法，即对测量值中包含的 NLOS 误差先鉴别后消除。

其中，间接法常用的是 Wylie 法。该算法首先对一段时间内的 TOA 测量值进行多项式拟合，然后利用 LOS 环境下测量值噪声标准差来判断测量值是否存在 NLOS 误差，再根据 TOA 测量值的残差进行 LOS 重构。由于这种方法在 LOS 与 NLOS 的混合环境下性能不佳，且题目中并没有 LOS 环境下噪声数据，因此不能选用 Wylie 法^[1]。

直接法中 CHAN 算法是 TOA 估计定位算法。该算法采用二次估计方法实现对终端位置的估计。首先利用坐标平移的方法，实现了对手持终端位置的一次快速估计。通过对一次评估结果误差的分析，得出了定位误差不仅与基站的几何位置有关，并且与基站与终端之间的距离有关。因此不同的基站对最终定位误差结果的“贡献”度不同，对不同的基站采用不同的权系数，从而提高定位精度。这种算法在二维平面下，对终端位置的定位准确度比较高而在三维中定位并不准确。这是由于三维精确定位相较于二维定位的难度在于 Z 坐标的定位阈值范围要比 X 坐标和 Y 坐标小得多。Z 坐标的很小的误差也有可能被 10 倍放大在 X、Y 轴上从而对定位精度产生严重影响^[2]。

由于 CHAN 算法在三维精确定位中的性能并不理想，考虑使用基于泰勒迭代的室内三维精确定位 TDOA/泰勒协同算法。TDOA 算法的优点在于不要求基站与手持终端间保持严格的时钟同步，甚至不需要各个基站间的时钟同步。TDOA/泰勒协同算法的主要思想是，根据 TDOA 算法移动终端到从基站与到主基站之间的距离差，得出关于距离差的非线性方程组，使用泰勒级数迭代算法，并将 TDOA 计算出来的最小二乘解作为迭代算法的初值进行迭代运算，经过多次迭代，当迭代结果与前次迭代结果的差值小于某一阈值时停止迭代，并将此结果作为移动终端的计算坐标^[3]，流程图如图 4.1 所示。

但是通过 MATLAB 实现发现 TDOA/泰勒算法在 NLOS 与 LOS 混合的情况下，对手持终端的定位情况并不理想，这是因为此算法要求初始估计与真实位置接近，以保证算法的收敛性。我们在最小二乘法的基础上，提出基于最小二乘法的改进算法，确定移动终端的位置。

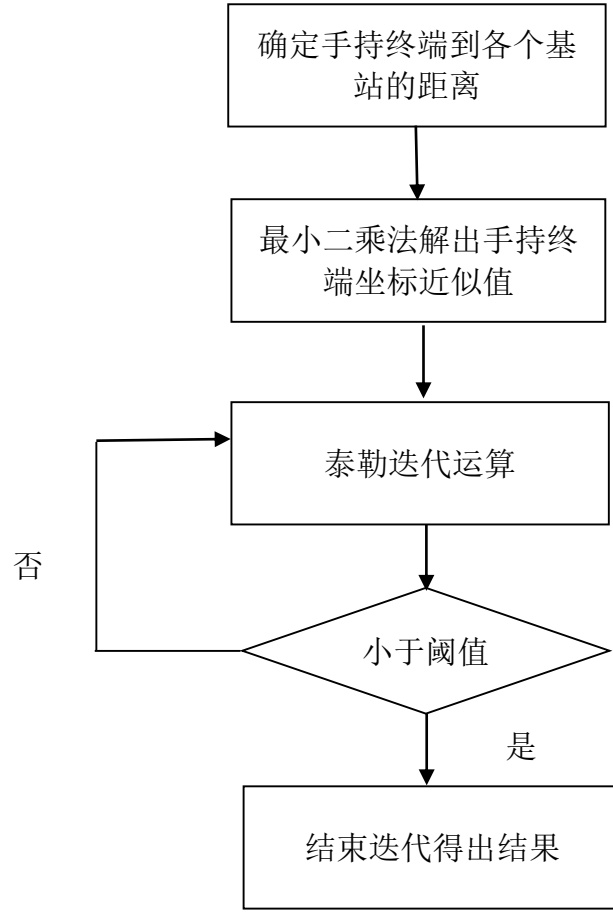


图 4.1 TOF/TDOA 协同算法流程图

4.3 模型建立

在无线定位系统中，最小二乘法是对移动终端定位使用较为广泛的方法之一。考虑到实际生活中，移动终端与基站的二维平面距离远大于垂直高度距离，本文首先从二维角度进行分析。

设移动终端的坐标为 (x, y) ， k 个基站坐标分别为 $(x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)$ ，则有

$$\begin{cases} r_i^2 = (x - x_i)^2 + (y - y_i)^2 \\ \bar{r}_i = ct_i \end{cases}, i = 1, 2, \dots, k \quad (4.1)$$

式 (4.1) 中， c 为电波传播速度， \bar{r}_i 是通过测量时间得到的基站与移动终端之间的距离， r_i 是两者间的真实距离。即在 LOS 路径传播下，移动终端必定位于以 (x_i, y_i) 为圆心，以 ct_i 为半径的 k 个圆上。然而在实际生活中，大部分的传播是

NLOS 路径, 因此 t_i 通常有一定误差, 无法直接求出各圆的交点, 确定移动终端的位置。根据测量值和已知量建立线性方程组:

$$AX = b \quad (4.2)$$

$$\text{其中, } A = -2 \times \begin{bmatrix} (x_1 - x_k) & (y_1 - y_k) \\ (x_2 - x_k) & (y_2 - y_k) \\ \vdots & \vdots \\ (x_{k-1} - x_k) & (y_{k-1} - y_k) \end{bmatrix}, \quad b = \begin{bmatrix} \bar{r}_1^2 - \bar{r}_k^2 - x_1^2 + x_k^2 - y_1^2 + y_k^2 \\ \bar{r}_2^2 - \bar{r}_k^2 - x_2^2 + x_k^2 - y_2^2 + y_k^2 \\ \vdots \\ \bar{r}_{k-1}^2 - \bar{r}_k^2 - x_{k-1}^2 + x_k^2 - y_{k-1}^2 + y_k^2 \end{bmatrix},$$

$$X = \begin{bmatrix} x \\ y \end{bmatrix}。$$

考虑到 NLOS 路径传播, 时间差等一系列误差, 合理的线性模型如下:
 $AX + N = b$ 。其中, N 为 $k-1$ 维随机误差向量。利用最小二乘原理, X 的解应该使模型误差 $N = b - AX$ 达到最小, 即令目标函数 $Q(X) = \|b - AX\|^2$ 最小。对 $Q(X)$ 关于 X 求导并令其等于 0, 求得 X 的最小二乘估计:

$$X_{LS} = (A^T A)^{-1} A^T b。$$

最小二乘法在 NLOS 环境下, 误差很大。在文献[4]中, James J Caffery 提出了 WLS 算法。该方法在最小二乘法的基础上, 利用残差函数的方式对其进行了改进。设移动终端的坐标为 (x, y) , k 个基站坐标为

$(x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)$, 其残差函数为:

$$f_i(x) = ct_i - \sqrt{(x_i - x)^2 + (y_i - y)^2} \quad (4.3)$$

残差平方和函数为:

$$F(x) = \sum_{i=1}^k \alpha_i^2 f_i^2(x) \quad (4.4)$$

其中 α_i 为第 i 个 TOA 的权值, 对于 LOS 环境 α_i 取较大值, 对于 NLOS 环境 α_i 取较小值, 则可以最大程度降低 NLOS 的误差。

本文在 WLS 算法的基础上, 对其进行改进, 应用到三维空间的定位中。首先考虑移动终端与基站的水平距离和垂直高度相差较大, 本文选用不同的权值来减少误差所带来的影响。设移动终端的坐标为 (x, y, z) , k 个基站坐标分别为

$(x_1, y_1, z_1), (x_2, y_2, z_2), \dots, (x_k, y_k, z_k)$, 其残差函数为:

$$f_i(x) = \alpha[(x_i - x)^2 + (y_i - y)^2] + \beta(z_i - z)^2 - \bar{r}_i^2 \quad (4.5)$$

残差平方和函数为：

$$F(x) = \sum_{i=1}^k f_i^2(x) \quad (4.6)$$

其中， α 和 β 表示权值，由于二维平面上移动终端与基站的距离较大，而在高度上的距离相对小很多，因此 α 赋予较大的权值， β 赋予较小的权值，具体数值通过多组数据试错得到。 $\bar{r}_i = ct_i$ ，为移动终端到基站的测量距离。在 LOS 环境下，移动终端位于以 (x_i, y_i, z_i) 为球心， \bar{r}_i 为半径的球上。而在 NLOS 环境中，由于存在传播误差， k 个球并不会相交于同一点，因此将目标函数 $F(x)$ 求得最小值，可以确定移动终端的位置。

4.4 模型评价

基于基站的三维定位问题所面临的主要难点就是消除各种误差对 TOA 测量数据的影响。因此本文所建立模型的核心思想就是减少利用 TOA 测量值所计算出的距离与真实距离之差。本文提出的三维空间定位算法，相比于目前广泛使用的 CHAN 方法具有精确度高、易实现等优点。通过对题目中所给基站位置坐标的观察发现 Z 轴的变化很小，一般不超过 2 米，因此此模型针对这一特点进行了考虑，提出了带有多个权值的模型。这样避免了 X、Y 轴上的权值对 Z 轴的数据的影响，大大提高了三维空间的定位精度。

4.5 对测量方式的改进

将本文中的模型用于测试用例，再将得出的结果与真实值进行比较后可以发现，Z 轴上的最大差值将近 1，虽然与 X 轴、Y 轴上的差值相比很小，但是由于 Z 轴变化也非常小，因此单独计算 Z 轴上的误差率会高达 70%。鉴于此种情况，可以考虑通过其他硬件方式提高终端对 Z 轴方向上定位的准确率。例如可以利用智能手机内置的气压传感器用于测量高度。目前，苹果、三星等品牌的智能手机已经利用了此方法对手机所处高度进行测量，结合适当的算法，可以较好地对手手机所处高度进行精确定位。

5. 问题二模型建立与分析

5.1 问题描述与分析

在问题一中，我们使用了文档提供的所有基站坐标，完成了移动终端的定

位。理论上，确定一个移动终端的位置并没有必要使用所有基站坐标，甚至某些终端经过较多的折射反射，反而会对最终结果产生误差。为了节约时间成本、测量成本，我们希望能够最大程度的减少基站的数量，使用尽可能少的基站数量，近似最优的定位。

在三维空间中，定位一个点的精确位置至少需要四个球，这是在四个球交于一点的理想情况下，即 LOS 环境。但是，由于是室内三维定位的场景，信号的传播路径大多是 NLOS 环境，四个球定位一个点必然会产生较大的误差。因此，我们需要考虑在基站最少的情况下，达到近似最优的定位。在 NLOS 环境下，测量距离会大于实际距离，即移动终端出现在几个球的相交区域。因此，除了考虑基站的最少数量，还需要考虑基站的分布情况，令几个球的重合区域达到最小，提高定位的精度。

5.2 模型建立

首先以二维空间坐标为例，在 LOS 传播环境下，移动终端必定位于以 (x_i, y_i) 为圆心，以 ct_i 为半径的圆上，我们称之为定位圆。在 NLOS 传播环境下，由于 t_i 一般为正偏置，移动终端必然位于定位圆内部，也就是说，三个定位圆的相交区域近似可以确定移动终端的范围。由图 5.1 可知， t_i 越大，圆的相交区域越大，则移动终端的位置误差越大。因此，我们需要尽可能减小定位圆的相交区域，选择 TOA 较小的基站作为定位圆的圆心。我们采用基于三边测量的节点选择算法对基站进行选择，以实现近似最优的三维定位精度。

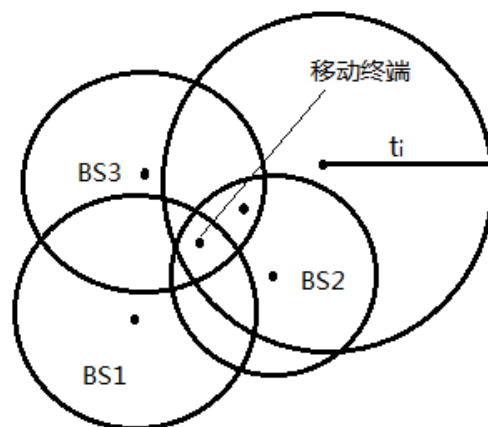


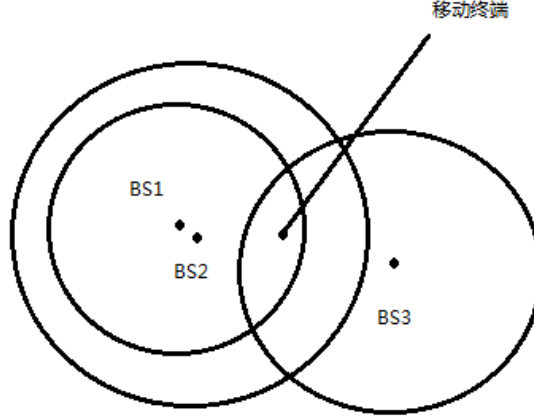
图 5.1 移动终端与基站的距离示意图

同理，在三维空间中，我们的目标是尽可能减少定位球的重叠区域。基于以上目标，我们可以对数据进行预处理。大体遵循以下原则：

- (1) 至少保留四个基站的 TOA 测量值用于定位移动终端位置；
- (2) 同等条件下优先保留小的 TOA 测量值；
- (3) 若出现一个或多个定位球被某个定位球完全覆盖的情况，则可剔除覆盖范围较大者，如图 5.2 所示，可以剔除 BS2（图 5.2 以二维空间为例，同样

适用于三维空间)；

(4) 若出现某个定位球被其他的所有定位球完全覆盖的情况，则所有的 TOA 测量值都保留。



5.2 定位圆完全覆盖的情况

基于以上原则，对基站的选择流程如下：

(1) 构造矩阵 $A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k1} & a_{k2} & \cdots & a_{kk} \end{bmatrix}$ ， $a_{ij} = t_i - t_j$ ，其中， i, j 表示基站的序号， k 表示基站总数；

(2) 构造函数 $B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1k} \\ b_{21} & b_{22} & \cdots & b_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ b_{k1} & b_{k2} & \cdots & b_{kk} \end{bmatrix}$ ， b_{ij} 表示基站 i 和基站 j 之间的距离

所对应的电波传播时间，取 $b_{ii} = 1$ ；

(3) 记录矩阵 $A-B$ 中大于零的元素所在行、列，形成一个 $n \times 2$ 的矩阵 C ，每行对应一对有覆盖的定位圆，第一列是覆盖者，第二列是被覆盖者；

(4) 统计第二列共有多少个不同元素，每个元素各有几个，若存在某个或某几个元素的个数大于 3，则删除矩阵 C 中该元素所在的行；

(5) 统计新的 C 中第一列有多少个不同元素，每个元素各有几个，若不同元素只有 4 个或更少，则可剔除它们对应的 TOA 数据，剩余的 3 个(或更多)TOA 数据是有效数据；

(6) 若上一步统计的不同元素有五个以上，则首先从中挑出个数大于 1 者，并剔除对应 TOA 数据，然后按从大到小的顺序剔除剩余元素所对应的 TOA 测量值，保证最终剩余 4 个有效的 TOA 测量值；

(7) 利用筛选出来的基站对移动终端进行定位。

理论而言，在三维空间坐标系中，四个定位球相交于一点可以确定移动终端的位置。但是考虑到室内定位的场景，本文最终选定了六个定位球用于目标函数的计算。

最终选取的六个用于定位的基站具有以下特点：其中两个基站为半径最小的球的球心，另外四个基站为半径相近的球的球心，近似分布于一个正三棱锥上。以正三棱锥顶点形成的四个球能够较为准确的定位一个点，但是室内定位的场景需要考虑 NLOS 的传播环境，因此本文又选择了两个接近 LOS 传播的基站作为约束条件，提高定位的准确性。接下来就证明以正三棱锥顶点形成的四个球能够较为准确的定位一个点。

首先从二维空间进行论证，本文可以证明当三个基站位于等边三角形上时，定位误差最小^[5]。假设在二维空间中存在三个节点 $p_i = (x_i, y_i), i=1,2,3$ ，我们需要通过 r_i (p_i 和 p 之间的距离) 来计算未知节点 p 的坐标 (x_0, y_0) ，计算公式如下：

$$\begin{cases} (x-x_1)^2 + (y-y_1)^2 = r_1^2 \\ (x-x_2)^2 + (y-y_2)^2 = r_2^2 \\ (x-x_3)^2 + (y-y_3)^2 = r_3^2 \end{cases} \quad (5.1)$$

通过求解等式 (5.1)，可以得到 p 的坐标

$$\begin{cases} x_0 = \frac{1}{\Delta} (2T_1(y_1 - y_3) - 2T_2(y_1 - y_2)) \\ y_0 = \frac{1}{\Delta} (2T_2(x_1 - x_3) - 2T_1(x_1 - x_2)) \end{cases} \quad (5.2)$$

其中， $T_1 = r_2^2 - r_1^2 - x_2^2 + x_1^2 - y_2^2 + y_1^2$ ， $T_2 = r_3^2 - r_1^2 - x_3^2 + x_1^2 - y_3^2 + y_1^2$ ，

$$\Delta = 4((x_1 - x_2)(y_1 - y_3) - (x_1 - x_3)(y_1 - y_2))$$

上述分析表明，一个未知的节点 p 可以由三个不在一条直线上的已知节点 p_1 ， p_2 和 p_3 确定。在实际应用中，可以将已知节点布局成锐角三角形来确保 Δ 相对较大，并且， p_1 ， p_2 和 p_3 不能太接近，否则 Δ 太小，无法准确定位未知节点的位置。显而易见，过于接近的三个点类似于一个点，无法很好的反映未知节点的位置信息。

上述分析中，本文假设测量距离是准确的，可是在实际场景中测量误差无可避免。以下部分本文假设误差范围在 $(-\varepsilon, \varepsilon)$ 之间，也就是说， p_i 和 p 之间的测量距离是 r_i ，实际距离在 $(r - \varepsilon, r + \varepsilon)$ 之间。

由于定位误差的存在，三个圆相交于一个区域，这个相交的区域被视作定位误差。在上述假设的前提下，等式 (5.1) 无法计算得到未知节点的准确位置。事实上在后续的分析中可以发现，未知节点的准确位置在等式 (5.1) 的解的位置附近。定义集合

$$C_{p_i} = \left\{ (x, y) \mid (x - x_i)^2 + (y - y_i)^2 \leq (r_i + \varepsilon_i)^2, (x - x_i)^2 + (y - y_i)^2 \geq (r_i - \varepsilon_i)^2 \right\}$$

当 $\varepsilon_i = 0$ 时, $\cap_i C_{p_i}$ 只包含一个点, 即等式 (5.1) 的解, 但是考虑到测量误差,

即 $\varepsilon_i > 0$, 因此 $\cap_i C_{p_i}$ 为一个区域。

用 S_p 表示圆 $\{(x, y) \mid (x - x_0)^2 + (y - y_0)^2 = \varepsilon^2\}$, 用 l_{p, p_i} 表示 p_i 和 p 之间的实际距离, l_{p, p_i} 与 S_p 相交于两点 $q_{i,1}$ 和 $q_{i,2}$, 用 $\tilde{l}_{q_{i,j}}$ 表示与 S_p 相交于 $q_{i,j}$ 的线段, 其中 $j=1,2$, 用 \tilde{C}_{p_i} 表示线段 $\tilde{l}_{q_{i,1}}$ 与 $\tilde{l}_{q_{i,2}}$ 之间的区域, 用 $\alpha_{i,j}$ 表示 $\overline{pp_i}$ 和 $\overline{pp_j}$ 之间的夹角。

当测量误差 ε 相对较小时, 在 p 的邻域中, C_{p_i} 可以被线性化并通过 \tilde{C}_{p_i} 来估计, 同时 C_{p_i} 的面也能够通过 \tilde{C}_{p_i} 的面积估计得到。可以证明, 当已知节点对称分布时, 可以得到最优的结构。

为了论证在二维空间中, 当三个已知节点位于等边三角形上时, 定位误差最小, 本文引入如下定义和引理。

定义 1. 子集 $S \subset R^n$ 中的点 $x_1 \in S$ 和 $x_2 \in S$, 对于任意 λ ($0 \leq \lambda \leq 1$), 都有 $\lambda x_1 + (1 - \lambda)x_2 \in S$, 则称 S 为凸子集。

定义 2. 对于任意点 $x_1 \in S$ 、 $x_2 \in S$ 和任意任意 λ ($0 \leq \lambda \leq 1$), 都有 $f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2)$, 则称 $f(x)$ 为凸函数。若满足 $f(\lambda x_1 + (1 - \lambda)x_2) < \lambda f(x_1) + (1 - \lambda)f(x_2)$, 则称 $f(x)$ 为严格凸函数。

引理 1. 对于定义在凸子集 $S \subset R^n$ 上的连续函数 $f(x)$, $f(x)$ 为凸函数当且仅当对于 S 中的所有 x , $f''(x) \geq 0$ 成立, $f(x)$ 为严格凸函数当且仅当对于 S 中的所有 x , $f''(x) > 0$ 成立。

引理 2. 对于定义在 S 上的凸函数 $f(x)$ 和 m 个点 p_1, p_2, \dots, p_m ,

$$f\left(\frac{1}{m}(p_1 + p_2 + \dots + p_m)\right) \leq \frac{1}{m}(f(p_1) + f(p_2) + \dots + f(p_m)), \text{ 如果 } f(x) \text{ 是严格凸函}$$

数, 等号成立当且仅当 $p_1 = p_2 = \dots = p_m$ 。

基于以上定义和引理，本文可以证明当 $\alpha_{1,2} = \alpha_{2,3} = \alpha_{3,1} = \pi/3$ 时， \tilde{C}_{p_i} 的面积达到最小。也就是说，对称的已知节点可以构成最优的结构。证明如下：
简单计算可得

$$area(\tilde{C}) = 2\varepsilon^2 \left(\tan \frac{\alpha_{1,2}}{2} + \tan \frac{\alpha_{2,3}}{2} + \tan \frac{\alpha_{3,1}}{2} \right)$$

其中， $\alpha_{1,2} + \alpha_{2,3} + \alpha_{3,1} = \pi$

由于当 $0 \leq x \leq \pi/2$ 时， $(\tan x)' = 2 \tan x(1 + \tan x) \geq 0$ ，可以得到

$$area(\tilde{C}) \geq 6\varepsilon^2 \tan \frac{\alpha_{1,2} + \alpha_{2,3} + \alpha_{3,1}}{6} = 6\varepsilon^2 \tan \frac{\pi}{6}$$

等号成立的条件为 $\alpha_{1,2} = \alpha_{2,3} = \alpha_{3,1} = \pi/3$ 。

由此可以得出结论，当已知节点对称分布时可以得到最优结构。将结论推广到三维空间，即四个已知节点对称分布于空间中，即形成正四面体时，可以得到最优结构。

5.3 数据分析

本文使用第一题给出的 5 组测试数据，对算法进行 MATLAB 实现，通过对实验数据和真实数据之间进行误差分析，来确定可以实现近似最优的三维定位精度的最少的基站数目。

我们是用信号在基站与终端之间的传播时间也就是 TOA 测量数据来对一个终端进行三维定位，因此基站的选择不同，会很大程度上影响终端三维定位的精度。根据基于三边测量的节点选择算法，我们选取两种基站来对终端进行三维定位。第一种为距离终端最近的一组基站记为 B1 型基站。第二种为距与终端距离相差不大的一组基站记为 B2 型基站。不断改变 B1 型与 B2 型基站的个数，使终端的三维定位精度达到最优，来确定最少需要的基站数目。

三维定位的误差分析，以第一组测试数据为例，我们主要通过计算 1000 组实际终端坐标与真实终端坐标的误差距离均值进行分析，公式如下：

$$Error = \frac{\sum_{i=0}^n \sqrt{(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2}}{n} \quad (5.3)$$

同时我们将终端的实验坐标与真实坐标之间的距离误差分为五级。第一级：误差在 0-5 米之间。第二级：误差在 5-10 米之间。第三级：误差在 10-20 米之间。第四级：误差在 20-40 米之间。第五级：误差在 40 米以上。通过分析 1000 组坐标误差在这五级之中的占比来进行误差分布分析。

首先，我们先将 B1 型基站定为 1 个，改变 B2 型基站的个数。误差数据如表 5.1 所示：

表 5.1 B2 型基站数目与误差关系表

B2 数目	1	2	3	4	5
Error	150.95m	127.08m	113.65m	68.43m	64.22m

根据上表可以看出随着 B2 型基站的个数的增多, 距离误差的均值逐步的下降, 当 B2 型基站增加到 5 个时, 距离误差均值的下降程度明显放缓。我们再观察距离误差在五级中的占比。

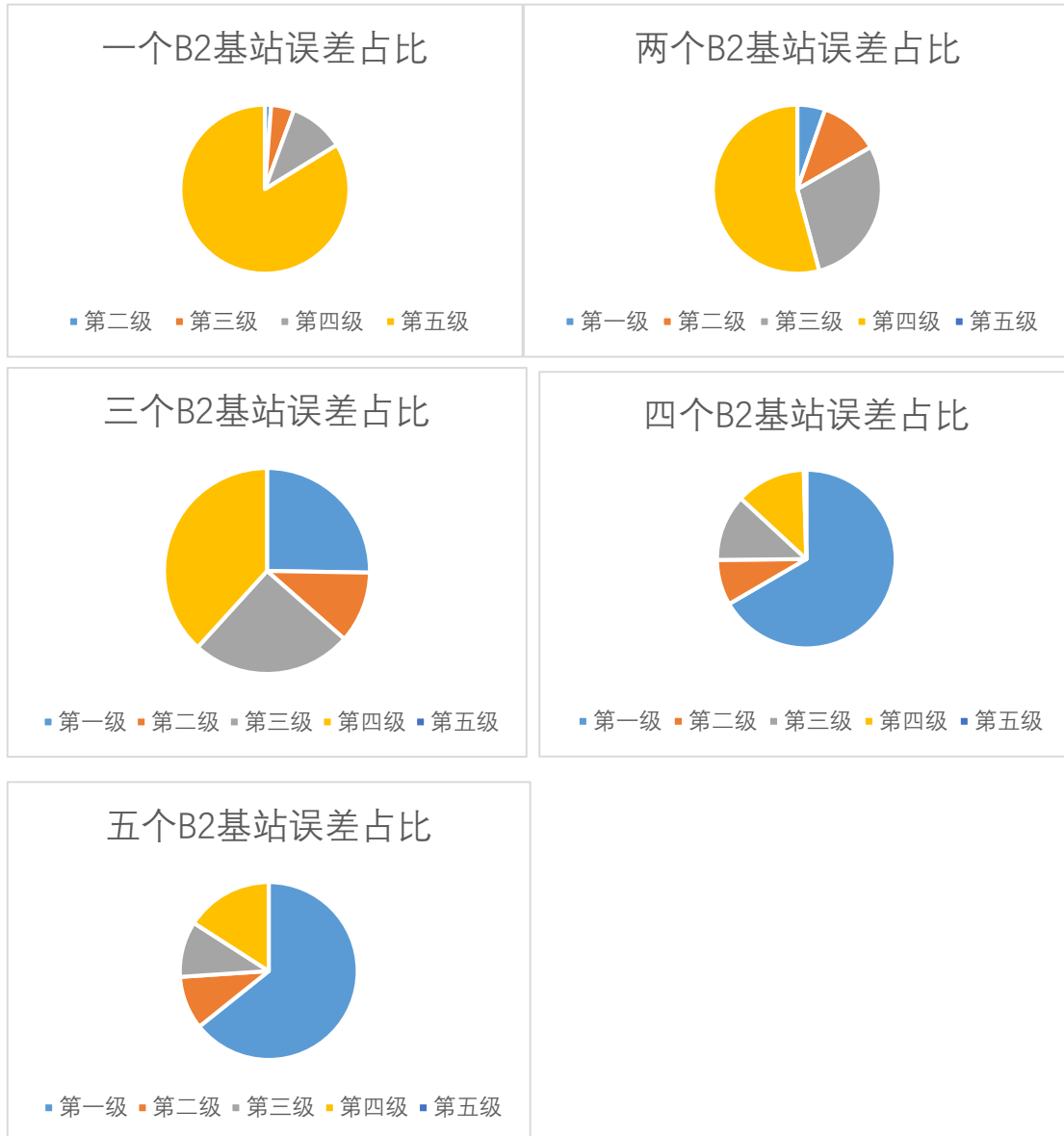


图 5.3 B2 基站误差占比图

可以从上面五个饼状图可以看出, B2 型基站增加到 4 个时, 1000 个样本点大部分分布在误差范围 0-5 米的第一级。再增加 B2 型基站, 对误差的下降, 效果不强。因此, 我们将 B2 型基站数目定为 4 个, 改变 B1 型基站的数目, 来进行误差分析。误差数据如表 5.2 所示:

表 5.2 B1 型基站数目与误差关系表

B1 个数	1	2	3	4
Error	68.43m	52.42m	48.6m	44.87m

当 B2 型基站为 4 个时，B1 型基站的个数越多，距离误差均值逐步下降，在 B1 个数为 2 时，误差均值下降明显放缓。我们再分析距离误差在五级误差中的占比。

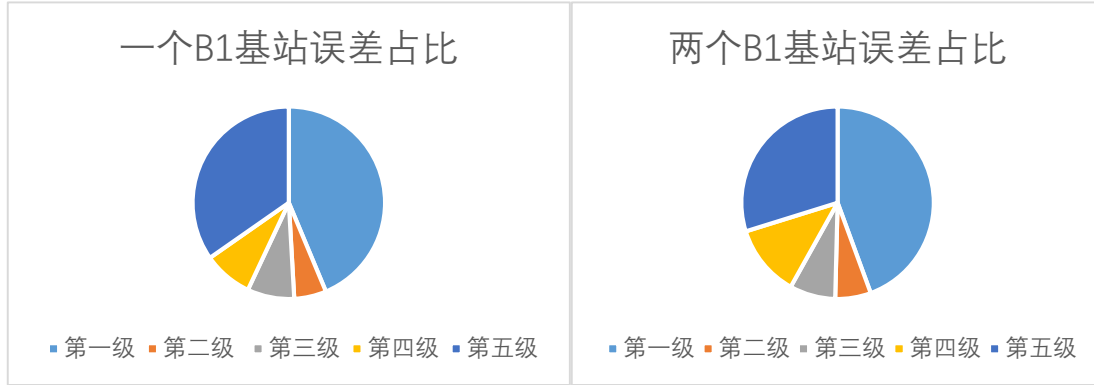


图 5.4 B2 基站误差占比图

可以从上面饼状图可以看出，当 B1 型基站数目增加对误差占比影响不大。经过综合考虑，我们认为最少 2 个 B1 型基站，4 个 B2 型基站，可以实现近似最优的三维定位精度。

6 问题三模型建立与求解

6.1 问题描述及分析

题目给定的移动过程中的终端采集到的 TOA 数据通过第一题所描述的算法可以得到终端的二维坐标数据。我们需要从这些组得出的、有限个数据点中，设法找出一条连续光滑的曲线，以确定模拟函数的所有值，进而得到所需要的终端运动轨迹模型。目前，寻求模拟函数常用的方法主要有差值和拟合两种。其中差值法是通过已知点来确定模拟函数的方法，这要求模拟函数必须严格通过所有已知点，因此适用于数据准确且数据量小的情况，否则可能造成局部较大的误差。而拟合方法不要求模拟函数必须通过所有已知点，它追求的是所有点到模拟函数达到某种误差指标的最小化，适用于数据有误差、数据量较大的情况。终端的二维坐标数据样本点有 1000 组，且不可避免的存在一些误差较大的样本点，故使用最小二乘法进行拟合，并在最小二乘法的基础上采用随机化方法精确去除误差较大的样本点。

6.2 模型建立

该问题中，我们只考虑终端长时间的，有序的运动轨迹，忽略在短时间内终

端无序的往复运动的情况来对模型进行简化。五组坐标样本点的分布情况如下图所示：

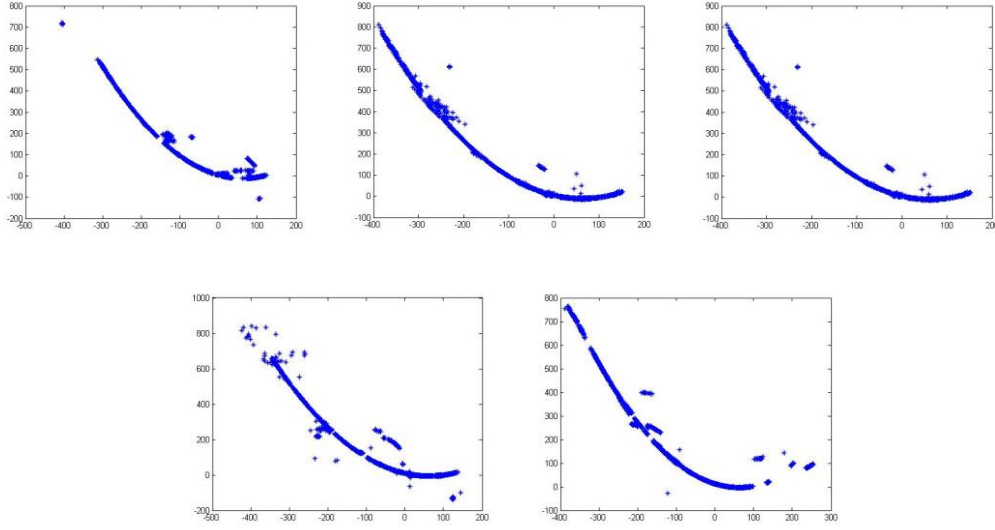


图 6.1 移动终端轨迹分布图

根据这些样本点的分布情况，我们可以将这些样本点用最小二乘法拟合成一条曲线。

(1) 最小二乘法曲线拟合方法

最小二乘法是在随机误差为正态分布时，由最大似然法推出的一个最优估计技术，它可以使测量误差的平方和最小。最小二乘技术主要是寻找参数集合，从而最小化数据点与曲线之间的距离度量。

一般形式的曲线方程如下：

$$Ax^2 + Bx + Cy + D = 0 \quad (6.1)$$

假设已知数据 $(x_i, y_i) (i=0, 1, \dots, m)$ ， Φ 为所有次数不超过 m 的多项式构成的函数

类，现求 $f_k(x_i) = \sum_{k=0}^n a_k x_i^k \in \Phi$ ，使得

$$I = \sum_{i=0}^m [f_k(x_i) - y_i]^2 = \sum_{i=0}^m \left(\sum_{k=0}^n a_k x_i^k - y_i \right)^2 \quad (6.2)$$

公式 (6.1) 为多元函数，其最小值存在的必要条件是其对应偏导等于零，由此可得，

$$\frac{\partial I}{\partial a_j} = 2 \sum_{i=0}^m \left(\sum_{k=0}^n a_k x_i^k - y_i \right) x_i^j = 0 \quad (6.3)$$

即

$$\sum_{k=0}^n \left(\sum_{i=0}^m x_i^{j+k} \right) a_k = \sum_{i=0}^m x_i^j y_i \quad (6.4)$$

公式(6.4)称为法方程,通过该方程可以求出唯一解 a_k 从而确定拟合函数。

(2) 随机化方法精确去除杂质点

直接最小二乘法并未考虑各样本点误差的差异,它假设样本数据是零均值的,且有共同的协方差阵,这与实际情况不符,因而导致参数的有偏估计,算法不令人满意。在此情况下,我们采用一种具有较高抗干扰能力的曲线拟合算法。

算法原理如下:

(a) 在所有样本点中随机选取 6 个样本点;

(b) 利用最小二乘法求解曲线参数 (A,B,C,D) ;

(c) 遍历所有样本点,求取各个点到已得到的曲线的最小距离,如果小于某一阈值,则称该样本点为匹配点,在数组 `index_inliers` 记录该样本点的编号,遍历完毕求取对于该拟合曲线的匹配点总个数 (`num_inliers`);

(d) 比较匹配点总个数(`num_inliers`)与匹配点个数的最大值(`max_inliers`)进行比较,当前者大于后者,将曲线参数 (A,B,C,D) 和记录匹配点编号的数组 (`index_inliers`) 保存在 `paras` 和 `index` 中,最后将 `num_inliers` 的值赋给 `max_inliers`;

(e) 循环执行步骤一到步骤四一定次数,调取曲线参数数据和匹配点的编号数据,就可以筛选出不匹配的样本点。

算法流程图如图 6.2 所示:

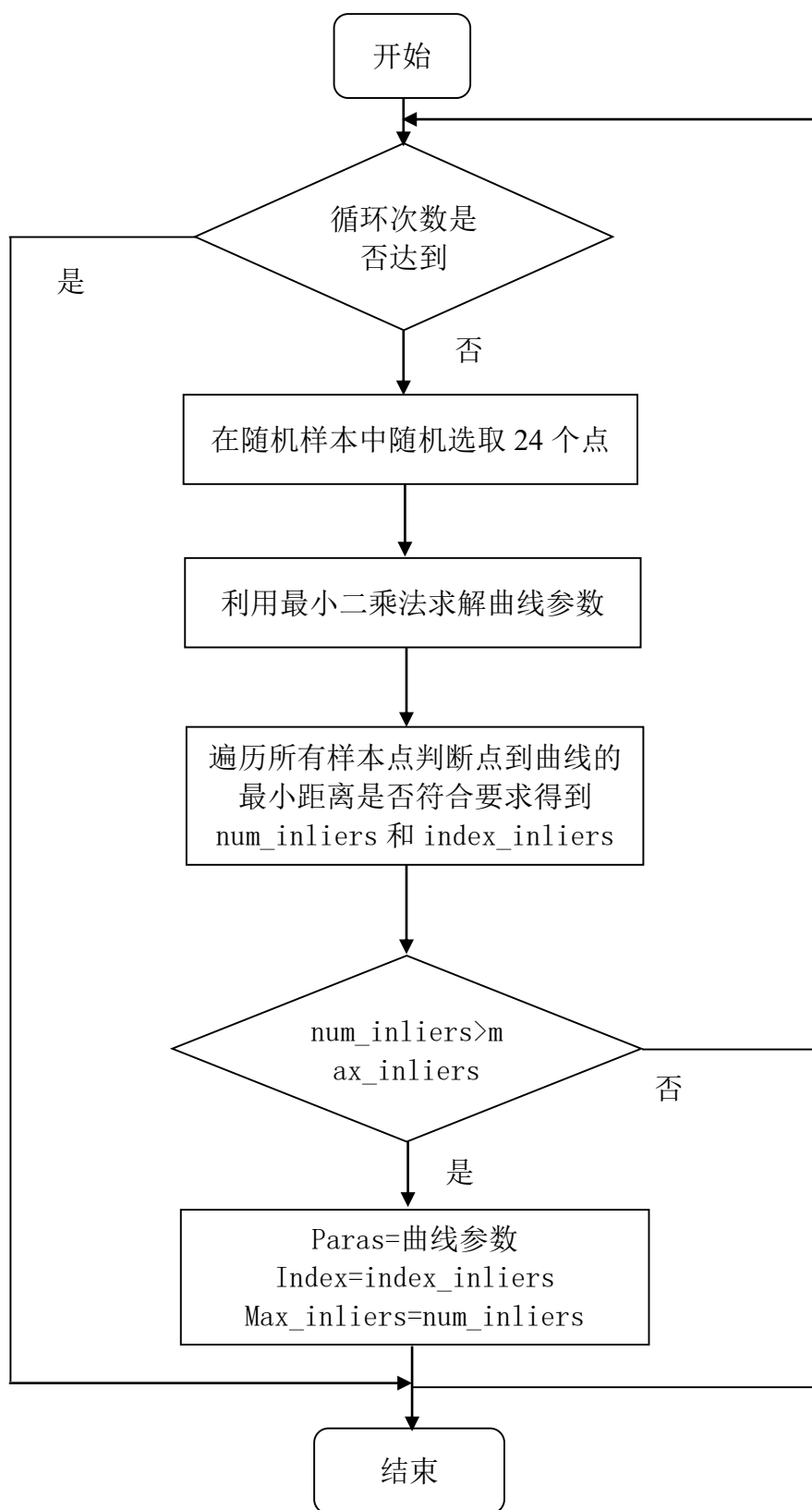


图 6.2 随机化方法流程图

6.3 实验数据分析

我们对五组移动终端轨迹建立如下函数模型： $f(x) = p_1x^2 + p_2x + p_3$ ，用上文提及的经过改进的最小二乘法分别计算出 p_1 、 p_2 和 p_3 的值，运行结果如下。

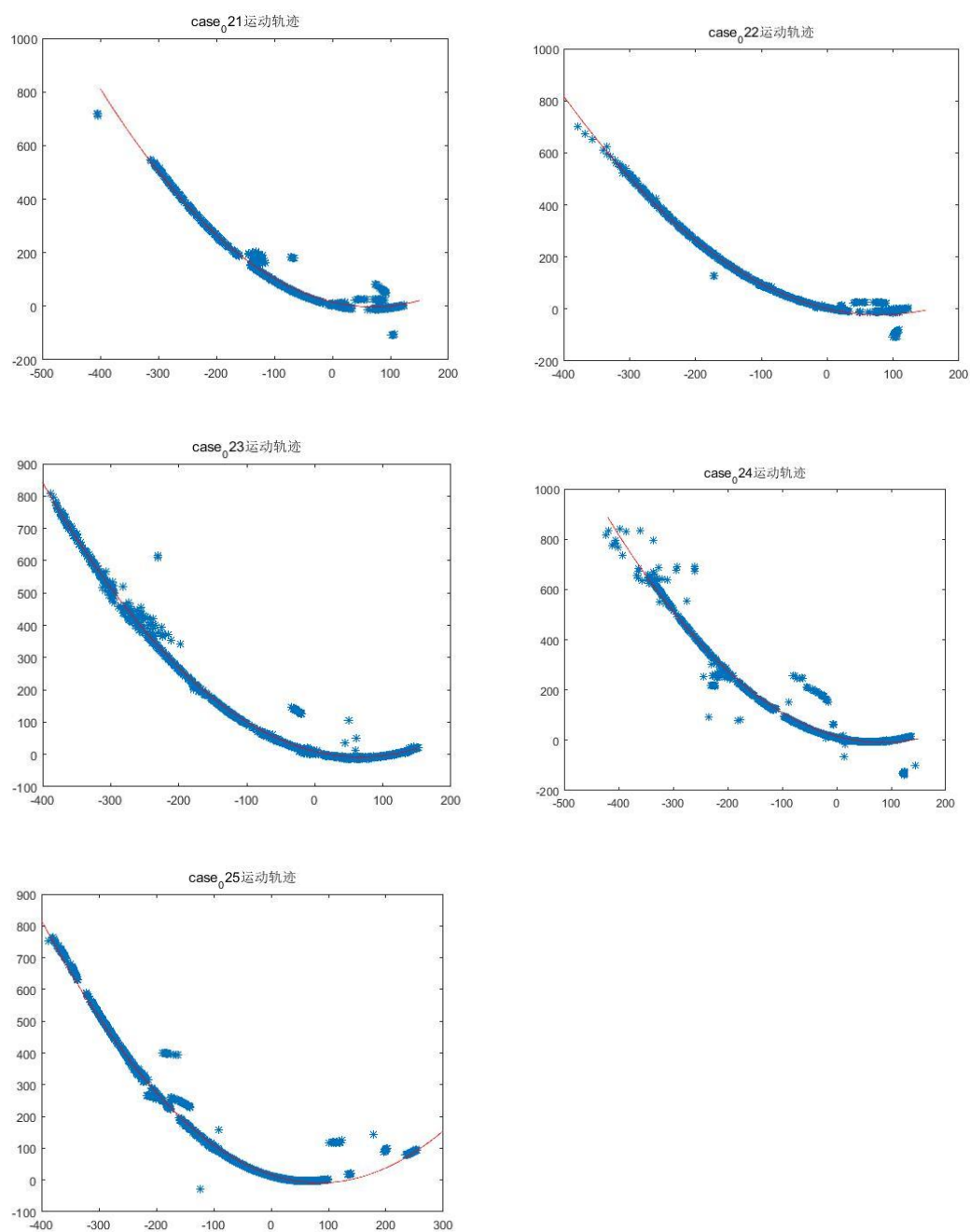


图 6.3 case₀21– case₀25 运动轨迹拟合结果图

第一幅拟合曲线的方程系数为 $p_1 = 0.003691$, $p_2 = -0.5165$, $p_3 = 15.37$, 第二幅

拟合曲线的方程系数为 $p_1 = 0.003639, p_2 = -0.5854, p_3 = 1.717$ ，第三幅拟合曲线的方程系数为 $p_1 = 0.003886, p_2 = -0.5236, p_3 = 9.743$ ，第四幅拟合曲线的方程系数为 $p_1 = 0.00352, p_2 = -0.5948, p_3 = 16.34$ ，第五幅拟合曲线的方程系数为 $p_1 = 0.003774, p_2 = -0.5453, p_3 = 19.08$ 。

针对曲线拟合的结果，我们选用 SSE, MSE, RMSE, R-square 四个常用指标对其进行评价。

(1) SSE(和方差)

该统计参数计算的是拟合数据和原始数据对应的误差的平方和，计算公式如下：

$$SSE = \sum_i^n w_i (y_i - y_i')^2 \quad (6.5)$$

SSE 越接近于 0，说明模型选择和拟合更好，数据预测也更成功。

(2) MSE(均方差)

该统计参数是预测数据和原始数据对应点误差的平方和的均值，也就是 SSE/n ，和 SSE 没有太大区别，计算公式如下：

$$MSE = \frac{SSE}{n} \quad (6.6)$$

(3) RMSE(均方根)

该统计参数，也叫回归系统的拟合标准差，是 MSE 的平方根，计算公式如下：

$$RMSE = \sqrt{MSE} \quad (6.7)$$

(4) R-square(确定系数)

该统计参数是通过数据变化来表征一个拟合的好坏，计算公式如下：

$$R-square = \frac{SSR}{SST} \quad (6.8)$$

该参数越接近 1，表明方程的变量对 y 的解释能力越强，这个模型对数据拟合的也比较好。

以第五组实验结果为例，曲线模型为：

$$f(x) = p_1 x^2 + p_2 x + p_3 \quad (6.9)$$

算法：最小二乘法

拟合曲线参数： $p_1 = 0.0043, p_2 = -0.3831, p_3 = 23.4536$

误差参数：

$$RMSE = 40.55$$

$$R-square = 0.9768$$

算法：改进的最小二乘法

拟合曲线参数: $p_1 = 0.003774, p_2 = -0.5453, p_3 = 19.08$

误差参数:

$$RMSE = 27.13$$

$$R-square = 0.9837$$

拟合效果如下:

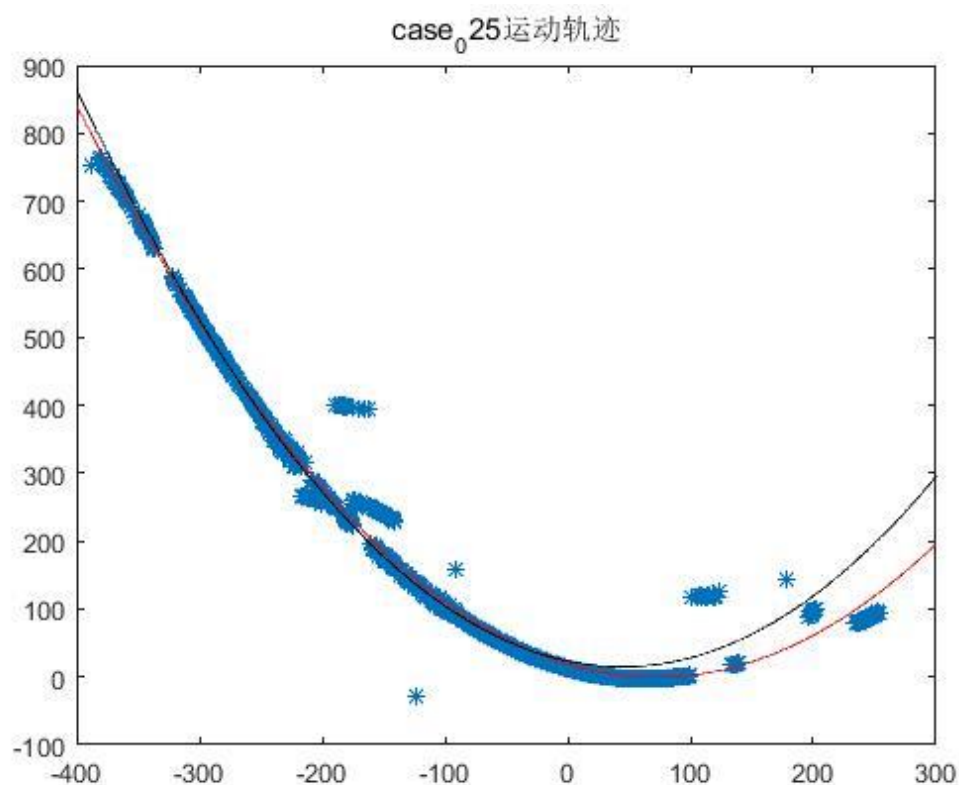


图 6.3 移动终端轨迹拟合图

图 6.3 中, 黑色曲线用最小二乘法曲线拟合, 红色曲线用改进后的最小二乘法进行曲线拟合, 可以定性的看出, 使用改进后的最小二乘法拟合更好。对误差指标进行定量分析, 改进后的算法拟合的均方根误差 (RMSE) 小于只用最小二乘法拟合的均方根误差 (RMSE), R-square 的值也更接近于 1, 对 y 的解释能力越强。

6.4 模型评价

(1) 模型优点

模型在求取最优曲线的过程中剔除了绝大多数的杂质点, 准确度得到提高的同时, 系统的鲁棒性也得到了增强。

(2) 模型缺点

算法所需要的运行时间较多, 不能适用于实时系统的要求。

7 问题四模型建立与求解

7.1 问题描述及分析

针对前面三个问题，本文所使用的模型都是基于同一个假设，即在给定的区域内，所有终端到每一个基站的距离都是可以通过 TOA 测量数据计算出来的。然而在实际情况中，基站的通信覆盖范围是有限的，只有当终端位于基站的通信半径内，才能测得准确的 TOA 数据。当终端与基站的距离超过该基站的通信半径时，虽然也能得到测量数据，但是应该认为此数据是无效的。同时，基于对问题二的讨论可知，在三维空间中，至少需要四个基站才能对一个终端进行较准确的定位，因此，针对具体的场景，需要计算出能够被此区域内基站定位的所有终端数量。

题目中还定义了终端的“平均连接度数”的概念，用 λ 表示：

$$\lambda = \frac{\text{所有可以被定位终端到基站之间的连接数}}{\text{终端数}} \quad (7.1)$$

由于每个场景中的终端数是一定的，而且也是已知的，因此，求解 λ 相当于求解分式中的分子部分。首先利用问题一的模型对五个场景(case026_input.txt 到 case030_input.txt)中的各个终端进行定位；然后再计算将每个终端与各基站之间的距离，如果距离小于等于 200 米，可以认为是有效定位；最后再统计有效定位的终端与基站的连接数目，即得到分子的数目。确定了各个场景的 λ 后，利用问题二所给出的判断定位精度的模型，对平均连接度数 λ 与定位精度之间的关系进行建模。

7.2 模型建立

对 case026_input.txt 到 case030_input.txt 中的数据进行观察可发现，各个基站的 Z 轴坐标范围比前三问中的数据要大，因此需要适当改变非线性规划中变量 Z 的最小值和最大值范围。此题中取每个场景中所有基站 Z 轴坐标的最小值作为变量 Z 的下界，Z 轴坐标的最大值作为上界。

修改相关参数后，利用问题一中的模型，分别对五个输入文件(case026_input.txt 到 case030_input.txt)中的所有终端进行定位，得到各个文件中所有终端的位置坐标（分别存放于 output_case_026.txt 到 output_case_030.txt 文件中）。

7.3 实验数据分析

下面以 output_case_030.txt 中的数据为例进行说明。

在此场景中，基站个数为 20，终端个数为 1000，为三维场景。output_case_030.txt 文件中共有 1000 组终端的三维坐标数据。根据三维空间中的距离坐标公式：

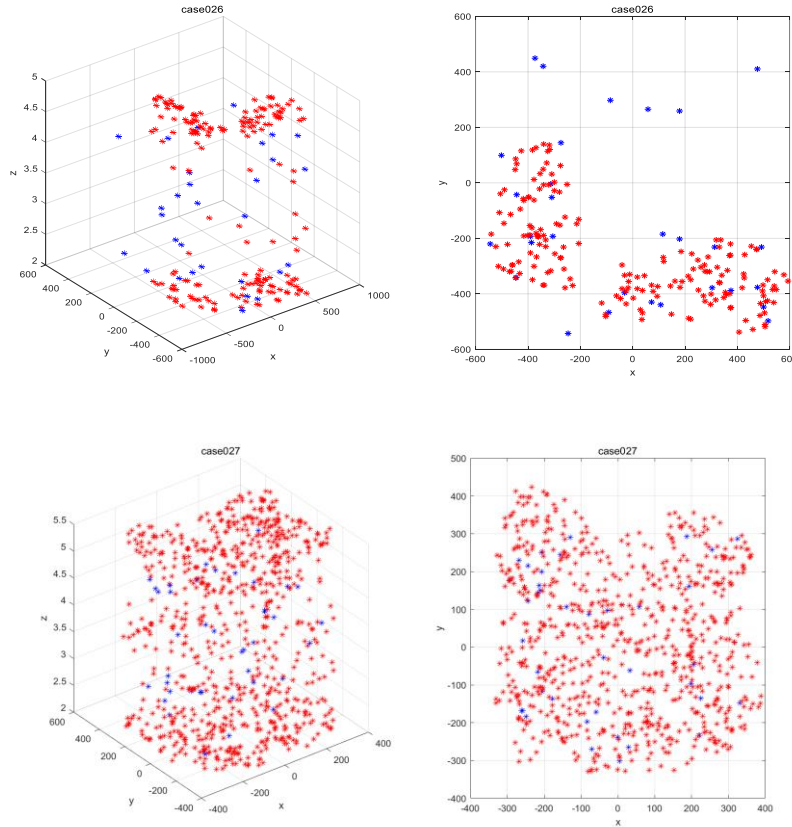
$$d_{(i,j)} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \quad (7.2)$$

其中, (x_i, y_i, z_i) 表示第 i 个终端的位置坐标 ($1 \leq i \leq 1000$), (x_j, y_j, z_j) 表示第 j 个基站的位置坐标 ($1 \leq j \leq 20$), $d_{(i,j)}$ 表示第 i 个终端到第 j 个基站的距离 (单位: 米)。

首先计算出第 1 个终端到 20 个基站的距离, 并统计出距离小于等于 200 米的个数, 以 counter 表示。如果 counter 等于或大于 4, 说明第 1 个终端可以被基站定位; 否则, 不能被定位。基于此规则, 再依次判断出其余各个终端能否被定位, 把能够定位的所有终端的 counter 值累加起来, 即为这个场景中的连接数。

按上述方法将其他场景的数据进行处理, 并将各类结果保存在 result_case026.txt 至 result_case030.txt 五个文件中 (分别对应 case026_input.txt 到 case030_input.txt 五中场景)。其中, 第 1 行第 1 列表示可被定位的终端数; 第 1 行第 2 列表示不能被定位的终端数; 第 2 行第 1 列表示连接数; 第 3 行第 1 列表示连接度数 λ ; 其余各行中, 第 1 列表示可被定位的终端标号, 第 2 列表示此终端的连接数。

为便于观察, 根据以上五个文件中的数据, 在 MATLAB 中绘出如下图所示的基站和可定位终端在三维空间中的分布图:



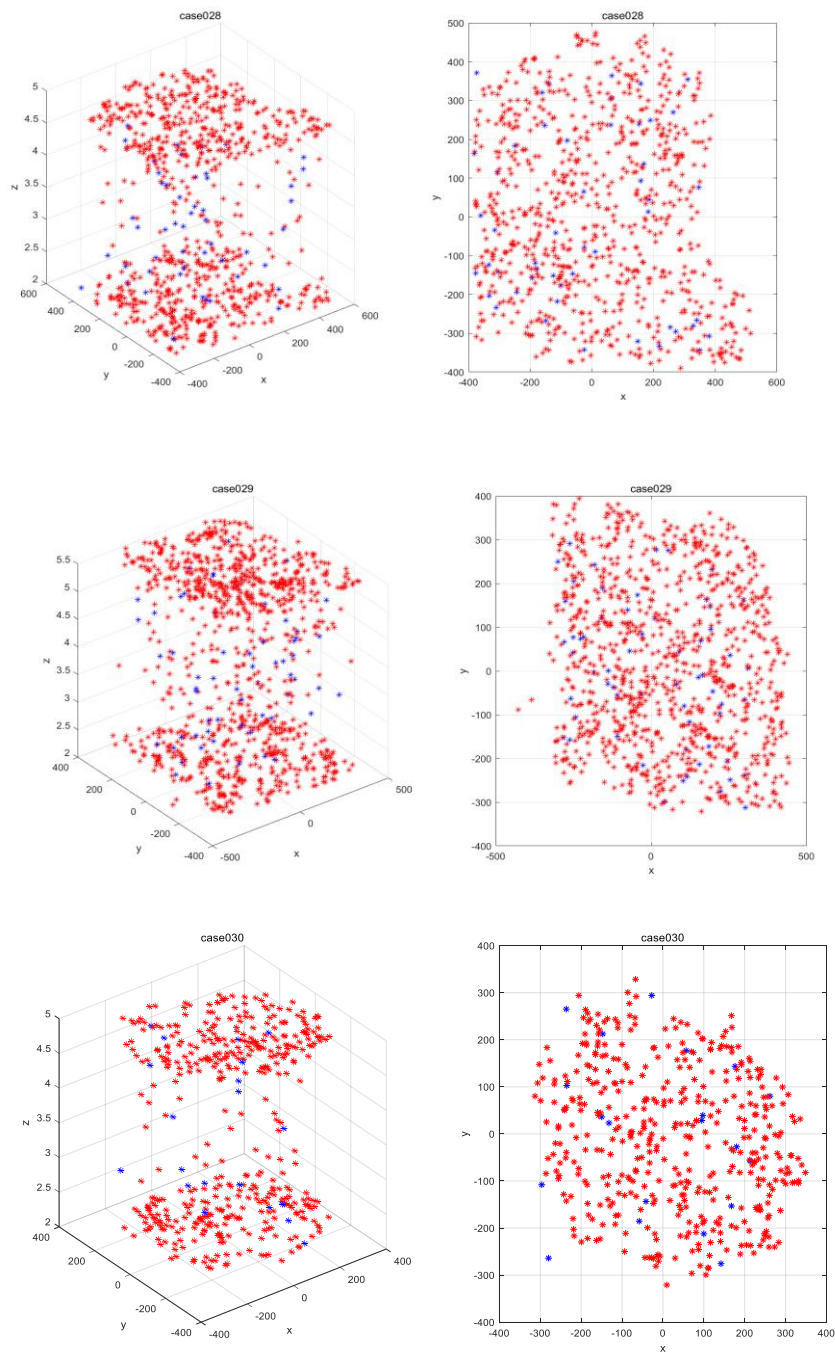


图 7.1 case_026-case_030 可定位终端与基站分布示意图

以上各图中，蓝色点表示基站的位置，红色点表示可定位终端的位置。左右两侧的图分别展示了在三维空间和二维空间中基站和可定位终端的相对分布情况。

根据“连接度数”的定义，分别计算出五个场景中的“连接度数”，结果如表 7.1 所示：

表 7.1 case_026-case_030 数据分析结果表

场景	λ	可被定位终端数/终端数	连接数
case026_input.txt	0.84	179/1100	921
case027_input.txt	5.58	766/1200	6701
case028_input.txt	5.07	779/1300	6592
case029_input.txt	9.45	978/1400	13224
case030_input.txt	2.52	438/1000	2515

将上表中的数据画成折线图，结果如下图所示：

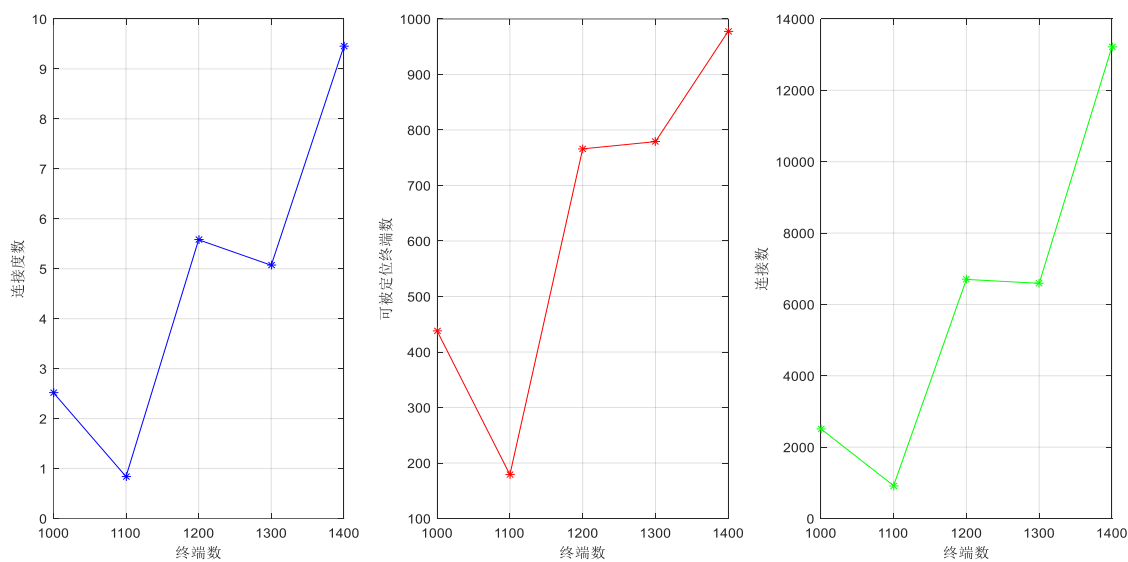


图 7.2 case_026-case_030 数据分析结果表

从图 7.2 可以看出，当终端数从 1000 个增加到 1400 个时，连接度数、可被定位终端数以及连接数的变化趋势基本一致。这三个指标的变化趋势与终端数的变化并不构成简单的线性关系。也就是说，在一个特定区域内，终端数量的多少，与有多少终端能够被定位、终端与基站的连接数目并没有直接的关系。结合图 7.1 可以发现，基站在空间中的分布情况对于终端的定位有很大的影响。

在 case026 这个场景中，基站的高度分布相差较大，在中间高度的基站数量较少，因此在三维空间的定位中，能够被定位的终端大多位于较低或较高的位置。如果只看 X 轴、Y 轴所构成的二维平面，可以看出在基站分布较密的区域，可被定位的终端较多；而基站分布较稀疏的区域，可被定位的终端很少甚至没有。

在其他四个场景中，基站在二维平面的分布比较均匀，因此可被定位的终端分布也比较均匀。而基站在 Z 轴方向上的分布都存在“两头多，中间少”的情况，因此可被定位终端的分布在 Z 轴方向上也是“两头多，中间少”。

通过以上的结果分析可知，在一个区域内，基站数和终端数的多少与可被定位终端数并不是简单的线性关系，而基站的空间分布则对其有重要的影响。

由问题二的结论可知，使用本文的模型得到近似最优的三维定位精度，需要至少 6 个基站，即当终端周围 200 米范围内至少有 6 个基站时，可以认为有很高的定位精度。因此，研究连接度数 λ 与定位精度的关系可以转换为研究连接度数 λ 与能够被精确定位的终端数之间的关系。

通过对result_case026.txt至result_case030.txt五个文件中的数据进行统计，可以得到所有可被定位终端中，终端到基站之间连接数至少为6个的终端数（用 α 表示）。所得数据统计在表中：

表7.2 case_026-case_030 α 与 λ 数据表

场景	λ	α
case026_input.txt	0.84	54
case027_input.txt	5.58	595
case028_input.txt	5.07	630
case029_input.txt	9.45	873
case030_input.txt	2.52	223

利用 MATLAB 曲线拟合工具箱^[8]，可以拟合出与 λ 之间的关系，如图 7.3 所示：

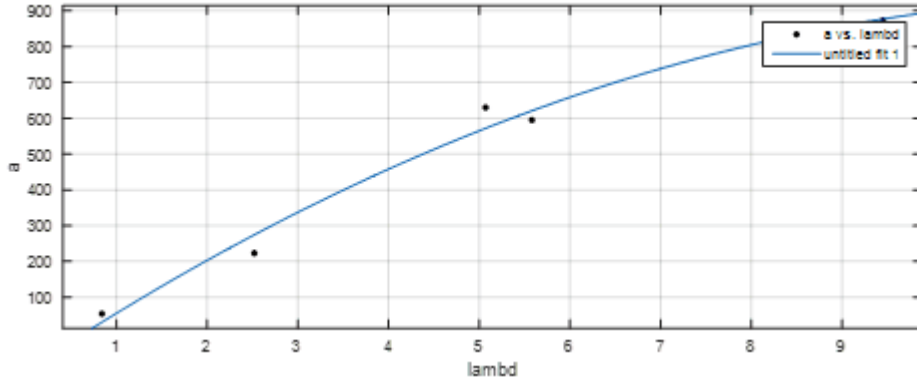


图 7.3 case_026-case_030 α 与 λ 关系示意图

α 与 λ 之间可以用二次函数近似表示，即

$$\alpha = p_1 \lambda^2 + p_2 \lambda + p_3 \quad (7.3)$$

其中， $p_1 = -6.777$ ， $p_2 = 168.1$ ， $p_3 = -106$ 。

以上分析说明连接度数与定位精度 λ 之间的关系可以用二次函数近似表示，因此随着连接度数 λ 的增加，定位精度提高的速率越来越慢，当连接度数 λ 达到一定数值时，定位精度不会有太大的提高。

8. 总结

根据题设要求以及提供的原始数据，本文基本上完成了所有四个问题的建模、解算及分析，针对各问题给出了相应的解答和注释，简要总结如下：

(1) 对 TOA 测量数据中包含的 NLOS 误差、噪声误差、时钟不同步误差，采取加权的方式进行减小。同时对 Z 轴单独赋予一个权值来抑制 Z 轴误差对 X, Y 轴定位精度产生的影响。

(2) 确定能够实现近似最优的三维定位的基站数量，从理论上对其进行了

证明。并从实际场景出发,考虑了基站分布的因素,用数据对比的方式说明了最少基站数目进行三维定位的可靠性。

(3) 因为采集到的 TOA 数据中存在许多误差较大的样本点,我们使用随机化方法去除杂质点,提高拟合精度。

(4) 基于对问题一和问题二的讨论,计算了五个场景中各终端与各基站之间的距离,并判断出能够被定位的所有终端。根据题目所给出的定义,计算了各个场景中的 λ 的值,并利用转化的思想,得出了 λ 与定位精度之间的关系。

参考文献

- [1] 贺远华,黎洪生. 距离几何 TOA 无线定位算法[J]. 计算机工程与应用, 2010, 12:112-114.
- [2] 杨天池,金梁,程娟. 一种基于 TOA 定位的 CHAN 改进算法[J]. 电子学报, 2009, 04:819-822.
- [3] 赵光旭. 监狱服刑人员三维精确定位系统研究与设计[D]. 辽宁工业大学, 2014.
- [4] Cafferry J, Stuber G. Subscriber location in CDMA cellular system[J]. IEEE Trans on VT, 1998, 47(5):406-416.
- [5] Han G, Choi D, Lim W. A Novel Reference Node Selection Algorithm Based on Trilateration for Indoor Sensor Networks[C]. International Conference on Computer and Information Technology. 2007:1003-1008.
- [6] 陈良波,郑亚青. 基于最小二乘法的曲线拟合研究[J]. 无锡职业技术学院学报, 2012, 05:52-55.
- [7] 闫蓓,王斌,李媛. 基于最小二乘法的椭圆拟合改进算法[J]. 北京航空航天大学学报, 2008, 03:295-298.
- [8] 卓金武. MATLAB 在数学建模中的应用[M]. 北京:北京航空航天大学出版社, 11-14, 2014.

附录

附录一

问题一代码:

question1.m

```
function f = Question1_1;
```

```
clear all;
```

```
clc;
```

```
%根据需要修改文件路径及文件名
```

```
input_file = textread('D:\Question1\case001_input.txt');
```

```

base_num = input_file(1,1);
mobile_num = input_file(2,1);

%base 为 30 个基站坐标，time 为终端到基站的传播时间
base = input_file(4 : base_num + 3, 1:3);
time = input_file(base_num + 4 : base_num + mobile_num + 3, 1:base_num);
%answer = answer_file(1:1100, 1:3);

% 光的传播速度
global SPD;
SPD = 3e8;

%基站的 X 轴坐标矩阵
axis_x = base(:,1);
%基站的 Y 轴坐标矩阵
axis_y = base(:,2);
%基站的 Z 轴坐标矩阵
axis_z = base(:,3);

% figure
% plot(axis_x, axis_y, '*');
% hold on
% grid

%距离矩阵，第 i 行为第 i 个终端到第 N 个基站的距离
distance = SPD * time;

for row = 1:mobile_num
    row
    %第 row 行终端到基站距离（已排序）
    radia = distance(row,:);
    radia_n = sort(radia);

    axis_x_n = ones(base_num,1);
    axis_y_n = ones(base_num,1);
    axis_z_n = ones(base_num,1);

    for i = 1:base_num
        for j = 1:base_num
            if radia_n(1,i) == radia(1,j)
                axis_x_n(i,1) = axis_x(j,1);
                axis_y_n(i,1) = axis_y(j,1);
                axis_z_n(i,1) = axis_z(j,1);
            end
        end
    end
end

```

```

        end
    end

    x_lb = axis_x_n(1,1) - radia_n(1,1);
    y_lb = axis_y_n(1,1) - radia_n(1,1);
    z_lb = 1;

    x_ub = axis_x_n(1,1) + radia_n(1,1);
    y_ub = axis_y_n(1,1) + radia_n(1,1);
    z_ub = 2;

    [coordinate, value] = fmincon(@(x)obj_fun(input_file, base_num, mobile_num, row, x), [2; 3;
    x_ub; y_ub; z_ub], [], [], [], [], [0; 0; x_lb; y_lb; z_lb], [100; 1000; x_ub; y_ub; z_ub],
    @(x)nonlcon_fuc(input_file, base_num, mobile_num, row, x))

    x_result = coordinate(3,1);
    y_result = coordinate(4,1);
    z_result = coordinate(5,1);

    file_result =
    fopen('F:\Users\MichaelLee826\Documents\MATLAB\Question1\output_case_001.txt', 'a');
    fprintf(file_result, '%7.2f %7.2f %5.2f\n', x_result, y_result, z_result);
    fclose(file_result);

    end
end

obj_fun.m
function f = obj_fun(input_file, base_num, mobile_num, row, x);

%base 为 30 个基站坐标，time 为终端到基站的传播时间
base = input_file(4 : base_num + 3, 1:3);
time = input_file(base_num + 4 : base_num + mobile_num + 3, 1:base_num);

% 无线电信号的传播速度
SPD = 3e8;

%基站的 X 轴坐标矩阵
axis_x = base(:,1);
%基站的 Y 轴坐标矩阵
axis_y = base(:,2);
%基站的 Z 轴坐标矩阵
axis_z = base(:,3);

```

```

%距离矩阵，第 i 行为第 i 个终端到各个基站的距离
distance = SPD * time;

%第 row 个终端到基站距离
radia = distance(row,:);

%按照半径，由小到大排序
radia_n = sort(radia);

%排序后的给基站三维坐标
axis_x_n = ones(base_num,1);
axis_y_n = ones(base_num,1);
axis_z_n = ones(base_num,1);

for i = 1:base_num
    for j = 1:base_num
        if radia_n(1,i) == radia(1,j)
            axis_x_n(i,1) = axis_x(j,1);
            axis_y_n(i,1) = axis_y(j,1);
            axis_z_n(i,1) = axis_z(j,1);
        end
    end
end

%目标函数中 x(1)-x(5)所表示的变量
% a->x(1)
% b->x(2)
% x->x(3)
% y->x(4)
% z->x(5)

%f 为目标函数
f = 0;

%找出半径最接近的三个球
i = 0;
j = 0;
k = 0;

for num = 1:base_num - 2
    if ((radia_n(1, num + 1) - radia_n(1, num)) < 20) && ((radia_n(1, num + 2) - radia_n(1,
num + 1)) < 20)
        i = num;
        j = num + 1;
        k = num + 2;
    end
end

```



```

        break;
    else
        i = 2;
        j = 3;
        k = 4;
    end
end

R = [radia_n(1,i), radia_n(1,j), radia_n(1,k)];
X = [axis_x_n(i,1), axis_x_n(j,1), axis_x_n(k,1)];
Y = [axis_y_n(i,1), axis_y_n(j,1), axis_y_n(k,1)];
Z = [axis_z_n(i,1), axis_z_n(j,1), axis_z_n(k,1)];

%得到目标函数
for i = 1:3
    r = R(i);
    x1 = X(i);
    y1 = Y(i);
    z1 = Z(i);
    f = f - x(1) * ((x(3) - x1)^2 + (x(4) - y1)^2) - x(2) * (x(5) - z1)^2 + r^2;
end
end

nonlcon_fun.m
function [g,h] = nonlcon_fuc(input_file, base_num, mobile_num, row, x);

%base 为 30 个基站坐标， time 为终端到基站的 TOA
base = input_file(4 : base_num + 3, 1:3);
time = input_file(base_num + 4 : base_num + mobile_num + 3, 1:base_num);

% 无线电信号的传播速度
SPD = 3e8;

%基站的 X 轴坐标矩阵
axis_x = base(:,1);
%基站的 Y 轴坐标矩阵
axis_y = base(:,2);
%基站的 Z 轴坐标矩阵
axis_z = base(:,3);

%距离矩阵， 第 i 行为第 i 个终端到各个基站的距离
distance = SPD * time;

%第 row 个终端到基站距离

```

```

radia = distance(row,:);

%按照半径， 由小到大排序
radia_n = sort(radia);

%排序后的给基站三维坐标
axis_x_n = ones(base_num,1);
axis_y_n = ones(base_num,1);
axis_z_n = ones(base_num,1);

for i = 1:base_num
    for j = 1:base_num
        if radia_n(1,i) == radia(1,j)
            axis_x_n(i,1) = axis_x(j,1);
            axis_y_n(i,1) = axis_y(j,1);
            axis_z_n(i,1) = axis_z(j,1);
        end
    end
end

%半径最小球的的基站坐标
x_min1 = axis_x_n(1,1);
y_min1 = axis_y_n(1,1);
z_min1 = axis_z_n(1,1);

%半径次小球的的基站坐标
x_min2 = axis_x_n(2,1);
y_min2 = axis_y_n(2,1);
z_min2 = axis_z_n(2,1);

%最小的两个半径
r_min1 = radia_n(1,1);
r_min2 = radia_n(1,2);

i = 0;
j = 0;
k = 0;

%找出半径最接近的三个球
for num = 1:base_num - 2
    if ((radia_n(1, num + 1) - radia_n(1, num)) < 20) && ((radia_n(1, num + 2) - radia_n(1,
num + 1)) < 20)
        i = num;
        j = num + 1;
    end
end

```

```

        k = num + 2;
        break;
    else
        i = 2;
        j = 3;
        k = 4;
    end
end

R = [radia_n(1,i), radia_n(1,j), radia_n(1,k)];
X = [axis_x_n(i,1), axis_x_n(j,1), axis_x_n(k,1)];
Y = [axis_y_n(i,1), axis_y_n(j,1), axis_y_n(k,1)];
Z = [axis_z_n(i,1), axis_z_n(j,1), axis_z_n(k,1)];

%g 和 h 为非线性规划中的非线性约束条件
g=[(x(3) - x_min1)^2 + (x(4) - y_min1)^2 + (x(5) - z_min1)^2 - r_min1^2
    (x(3) - x_min2)^2 + (x(4) - y_min2)^2 + (x(5) - z_min2)^2 - r_min2^2
    - R(1)^2 + x(1) * ((x(3) - X(1))^2 + (x(4) - Y(1))^2 + x(2) * (x(5) - Z(1))^2
    - R(2)^2 + x(1) * ((x(3) - X(2))^2 + (x(4) - Y(2))^2 + x(2) * (x(5) - Z(2))^2
    - R(3)^2 + x(1) * ((x(3) - X(3))^2 + (x(4) - Y(3))^2 + x(2) * (x(5) - Z(3))^2
    ];

h=[];
end

附录二
问题二代码：
comparison.m
clc;
clear all

%根据需要修改文件路径及文件名
target = textread('D:\Question2\sample_case001_ans.txt');
result = textread('D:\Question2\sample_output_case_001.txt');

mobile_num = 1000;

%终端的真实坐标
x_target = target(:, 1);
y_target = target(:, 2);
z_target = target(:, 3);

%求得的坐标
x_result = result(:, 1);

```

```

y_result = result(:, 2);
z_result = result(:, 3);

D = zeros(mobile_num,1);
d = zeros(mobile_num,1);
sum = 0;
counter_5 = 0;
counter_10 = 0;
counter_20 = 0;
counter_40 = 0;
counter_other = 0;

%统计误差分布
for i = 1:mobile_num
    D(i,1) = (x_target(i,1) - x_result(i,1))^2 + (y_target(i,1) - y_result(i,1))^2 + (z_target(i,1) -
z_result(i,1))^2;
    d(i,1) = sqrt(D(i,1));
    test = d(i,1);

    if test <= 5
        counter_5 = counter_5 + 1;
    elseif test > 5 && test <=10
        counter_10 = counter_10 + 1;
    elseif test > 10 && test <= 20
        counter_20 = counter_20 + 1;
    elseif test > 20 && test <= 40
        counter_40 = counter_40 + 1;
    else
        counter_other = counter_other + 1;
    end

    sum = sum + d(i,1);
end
d = sort(d);

%误差的平均值
aver = sum / mobile_num

s = 0;
for j = 1:mobile_num
    s = s + (d(j,1) - aver)^2;
end

%方差

```

```

err = s / mobile_num

data = [counter_5, counter_10, counter_20, counter_40, counter_other]
percent = data / mobile_num * 100

%绘制饼图
pie(percent)

```

附录三

问题三代码：

readdate.m

%根据需要修改文件路径及文件名

```

result1 = textread(' D:\Question3\拟合\output_case_021.txt');
result2 = textread(' D:\Question3\拟合\output_case_022.txt');
result3 = textread(' D:\Question3\拟合\output_case_023.txt');
result4 = textread(' D:\Question3\拟合\output_case_024.txt');
result5 = textread(' D:\Question3\拟合 output_case_025.txt');

```

```

x1 = result1(:,1);
y1 = result1(:,2);
x2 = result2(:,1);
y2 = result2(:,2);
x3 = result3(:,1);
y3 = result3(:,2);
x4 = result4(:,1);
y4 = result4(:,2);
x5 = result5(:,1);
y5 = result5(:,2);

```

```

figure;
plot(x1, y1, '*');
figure;
plot(x2, y2, '*');
figure;
plot(x3, y3, '*');
figure;
plot(x4, y4, '*');
figure;
plot(x5, y5, '*');

```

```

num = 1000;%循环次数
for i=1:num
    a = unidrnd(1000,1,24);
    b = sortrows(a');

```

```

num_inliers = 0;%符合要求的点的数目
max_inliers = 0;%拥有的最大数目
index = b';%随机的 24 个值
index_inliers = zeros(1,24);
x = zeros(1,24); %随机选取的 24 个点
y = zeros(1,24);
for j=1:24
    x(1,j) = x5(index(1,j),1);
    y(1,j) = y5(index(1,j),1);
end
paras = polyfit(x,y,2);
max_paras = zeros(1,3);
a1 = paras(1,1);
a2 = paras(1,2);
a3 = paras(1,3);

for m=1:1000
    x_d = -600:0.1:300;
    f = a1*x_d.^2+a2*x_d+a3;
    for k=1:length(x_d);
        den(k) = sqrt((x_d(k)-x5(m,1))^2+(f(k)-y5(m,1))^2);
    end
    erro = min(den);
    if erro<10
        num_inliers = num_inliers+1;
        %index_inliers = index;
    end
end
if num_inliers>max_inliers
    max_inliers = num_inliers;
    max_paras = paras;
    index_inliers = index;
else
    break
end
end
end

```

附录四

问题四代码：

counter.m

clc

clear all

%根据需要修改文件路径及文件名

```
mobile_located = textread('D:\Question4\result_case026.txt');
```

```
%可被定位的终端数
```

```
located_num = mobile_located(1, 1)
```

```
%所有可被定位的终端标号及其连接数矩阵
```

```
num = mobile_located(4:located_num + 3, 2);
```

```
%统计连接数大于等于 6 个的终端
```

```
counter = 0;
```

```
for i = 1:located_num
```

```
    if num(i) >= 6
```

```
        counter = counter + 1;
```

```
    end
```

```
end
```

```
%将结果写入文件
```

```
%根据需要修改文件路径及文件名
```

```
file_result = fopen('D:\Question4\counter_case030.txt', 'a');
```

```
fprintf(file_result, '%d\n', counter);
```

```
fclose(file_result);
```

```
draw.m
```

```
clear all;
```

```
clc;
```

```
%根据需要修改文件路径及文件名
```

```
input_file = textread('D:\Question4\case026_input.txt');
```

```
mobile_location = textread('D:\output_case_026.txt');
```

```
mobile_located = textread('D:\result_case026.txt');
```

```
%基站数
```

```
base_num = input_file(1,1);
```

```
%终端数
```

```
mobile_num = input_file(2,1);
```

```
base_location = input_file(4:base_num + 3, 1:3);
```

```
%基站坐标
```

```
x_base = base_location(:,1);
```

```
y_base = base_location(:,2);
```

```
z_base = base_location(:,3);
```

```
%终端坐标
```

```

x_mobile = mobile_location(:,1);
y_mobile = mobile_location(:,2);
z_mobile = mobile_location(:,3);

%可被定位的终端数
located_num = mobile_located(1, 1);

%绘制基站的三维空间分布图
figure
subplot(1,2,1)
plot3(x_base, y_base, z_base, 'b*');
title('case026');
xlabel('x');
ylabel('y');
zlabel('z');

hold on
grid

for i = 1:located_num
    for j = 1:mobile_num
        if j == mobile_located(i,1)
            %绘制终端的三维空间分布图
            plot3(x_mobile(j), y_mobile(j), z_mobile(j), 'r*');
        end
    end
end

%绘制基站的二维空间分布图
subplot(1,2,2)
plot(x_base, y_base, 'b*');
title('case026');
xlabel('x');
ylabel('y');

hold on
grid

for i = 1:located_num
    for j = 1:mobile_num
        if j == mobile_located(i,1)
            %绘制终端的二维空间分布图
            plot(x_mobile(j), y_mobile(j), 'r*');
        end
    end
end

```



```

        end
    end

draw_line.m
%绘制折线程序

clear all;
clc;

mobile_num = [1000, 1100, 1200, 1300, 1400];
lambd = [2.52, 0.84, 5.58, 5.07, 9.45];
located_num = [438, 179, 766, 779, 978];
connected_num = [2515, 921, 6701, 6592, 13224];

subplot(1,3,1);
plot(mobile_num, lambd,'b-*');
xlabel('终端数');
ylabel('连接度数');
grid

subplot(1,3,2)
plot(mobile_num, located_num,'r-*');
xlabel('终端数');
ylabel('可被定位终端数');
grid

subplot(1,3,3)
plot(mobile_num, connected_num,'g-*');
xlabel('终端数');
ylabel('连接数');
grid

fitting.m
%曲线拟合

clc;
clear;

lambd = [0.84 5.58 5.07 9.45 2.52];
a = [54 595 630 873 223];

cftool(a,lambd);

```