

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 重庆理工大学

参赛队号 11660022

	1.郭艳婷
队员姓名	2.史德容
	3.胡猛

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：

基于无线通信基站的终端定位对于无线网络的优化和信号追踪具有重要的意义。本文的主要工作是：基于 TOA 数据和基站坐标，通过时间线性模型和最小二乘法建立终端定位模型、用尽可能少的基站数定位终端坐标的模型、设计终端在移动过程中的轨迹求解算法以及分析终端的平均连接度数 λ 与定位精度的关系。

针对问题一，分析 case001-case010 的数据，因实际无线信号的多径传播，所给的 TOA 测量值必然大于或等于在 LOS 环境下的测量值。通过对 sample_case001-sample_case005 的模拟发现延时值（测量时间减去两点之间光直线传播的时间）随 TOA 测量值的增大而线性增加。为了定位终端坐标，首先建立延时值 \hat{t} 与 TOA 观测值 t 的线性模型 ($\hat{t} = kt + b$)，其次通过多个基站的几何位置及 TOA 测量值建立误差函数，然后利用泰勒级数展开方法进行近似计算使得误差函数最小，最终确定终端的位置坐标，具体结果见附件 1。通过运用 sample_case001_input-sample_case005_input 中的数据进行验证，发现本文的方法具有平均值误差在 3.5m 以内的良好结果。

针对问题二，分析 case010-case020 的数据，力求用最少的基站去定位终端，本文首先通过最小生成树算法，根据数据的分布动态选取初始聚类中心，然后利用 K-means 算法逐步找到定位终端的最少基站数。实现近似最优的三维定位中，我们求得最少基站数为 11 个基站。

针对问题三，通过 case021-case025 的数据，可得到移动终端在不同时刻

的位置坐标。为了描述运动轨迹，本文通过三次函数逼近终端的运动轨迹。终端的运动轨迹见图 15-19，观测这些图发现没有突变点和跳跃间断点，轨迹图像比较平稳。

针对问题四，通过问题 2 提出的聚类方法能找到定位一个终端所需的最少基站数目。根据每个基站通信半径覆盖范围以及基站与终端的距离可知终端的可通信的基站数，进而可知终端能否被定位。在一个网络中可被定义终端的平均连接度数 λ 与终端的定位精度呈负相关，具体结果见正文。

关键词：终端定位，最小二乘法，运动轨迹，定位精度

基于无线通信基站的室内三维定位问题

一、 问题重述

1.1 背景介绍

随着无线通信网络和移动互联网的蓬勃发展，提供基于地理位置信息的服务（Location Based Service，简称 LBS）已经成为最具市场前景和发展潜力的业务之一。从传统的 GPS 导航，到大众点评、微信等基于地理位置的消费信息服务和社交软件，实现其功能的基础就是要通过手机、导航仪等终端设备收发信号，来获得距离、角度等测量信息，并利用定位算法将这些测量信息转换成坐标信息。

基于无线移动通信网络的定位是以获取用户手持终端（包括手机或者平板等设备）的位置为目标。而达成这一目标的手段是通过测量无线电信号的强度、传播时间、到达角等物理指标，并将其转化成终端与基站之间的距离、角度等信息，最终利用定位算法将距离、角度等信息转化成终端的坐标信息。

相比于 GPS 等商用卫星定位系统，基于无线通信基站方式的定位可以规避很多问题。并且具有如下特殊性：

首先，通信基站的目标区域是 GPS 等卫星定位系统无法实现定位的场景。在高楼林立的城区，建筑物内部、地下停车场等区域，GPS 等系统是无法满足定位需求的。而这些应用场景基站、终端密集，是基站定位可以实现突破的地方。

其次，通信基站所处的电磁信号环境较之 GPS 等系统更加复杂。以室内环境为例，无线电信号的传播过程中会经过墙面的多次反射、室内物体的折射和吸收等。这些物理因素会导致通信基站测量得到的诸如距离、角度等信息存在噪声。如何基于这些有噪声的测量，得到对于位置信息的准确估计，也是通信基站实现对终端定位需要解决的问题。

基于通信基站的定位系统，将会成为羲和导航系统有力的技术手段。求解分析基站定位相关问题的创新性和可实现性强的算法，将有可能被快速部署到现代商业通信网络中，带来巨大的社会和经济效益。

1.2 本文研究的问题

无线电信号在大气中从 A 点向 B 点传播时，如果传播过程中存在一个没有遮挡的直达路径，那么，这种传播环境被称为视距传播环境（Line Of Sight

propagation, 简称 LOS)。这种传播环境如图 1 中的左图所示。如果在传播过程中, 由于建筑物或树木的遮挡、反射、折射等物理现象, 使得从 A 点到 B 点之间存在多条无线电信号的传播路径, 这种环境被称为非视距传播环境 (Non-LineOfSight, 简称 NLOS)。需要注意的是, 在 NLOS 传播环境中, 仍然可能存在着无线电波的直达路径, 只不过相比于 LOS 传播环境, 在 NLOS 环境下因为遮挡、反射和吸收等损耗, 信号强度会在传播过程中变得较弱。



图 1 LOS 径与非 LOS 径示意图

图片来源 <https://sites.google.com/site/sanaeeg473/wimax-architecture>

通信基站的定位环境十分复杂, 与周围的环境密切相关, 不同场景下基站与终端之间距离和实际值存在差异。本文具体解决以下四个问题:

问题 1: 在给定 10 组 LOS 或 NLOS 传播环境下从手持终端到基站的 TOA 测量数据和所有基站的三维坐标下 (对应附录中编号为 case001_input.txt 到 case010_input.txt 的文件), 采用数学建模方法, 建立模型, 计算出终端的三维坐标。

问题 2: 在给定 10 组 TOA 测量数据和所有基站的三维坐标下 (对应附录中编号为 case011_input.txt 到 case020_input.txt 的文件), 我们设计了一个算法, 使用相对较少的基站数目, 实现近似最优的三维定位精度。

问题 3: 在给定 5 组处于移动过程中的终端采集到的 TOA 数据下 (对应附录中编号为 case021_input.txt 到 case025_input.txt 的文件), 设计了一个算法计算出终端的运动轨迹。(此时, 编号为 case021_input.txt 到 case025_input.txt 的文件中, 只记录一个终端的 TOA 数据, 并且是这一个终端在运动轨迹中多个位置上的 TOA 数据。

问题 4: 在前述 3 问中, 都是假设给定区域内终端到每一个基站的距离都是可知的, 但事实上, 基站的通信半径是有限的, 因此, 只有在基站通信半径覆盖范围内的终端才有可能测到自身到基站的距离。而一个终端只有获得它与足够数目的基站之间的距离测量值, 才能完成定位。现今假设每个基站的通信半径为 200 米 (超过范围虽然有测量数据, 但无效)。我们根据给定的 5 组测量信息数据集 (对应附录中编号为 case026_input.txt 到 case030_input.txt 的文件) 设计算法寻找出被基站定位的所有终端。进一步, 对每一个场景中 (对应着

case026_input.txt 到 case030_input.txt 五个文件中的一个) 定义终端的平均“连接度数”为

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

建立模型，得到连接度数与定位精度之间的关系。

二、模型假设与符号说明

2.1 模型假设

- (1) 假设基站以球面波向全空间发射无线信号。
- (2) 假设在整个网络中包含时区误差引起的样本量非常小。
- (3) 假设基站发射信号稳定，每个发射信号完全相同。
- (4) 假设基站各个方向具有相同单位增益，能够接收来自全空间的无线信号。
- (5) 同一场景下，引起多径效应的重要散射体之间的相对位置在整个测试时间内不会发生巨变。
- (6) 同一场景下，主要多径效应相对于各个基站的时延抖动规律是相同的。

2.2 符号说明

符号	说明
(x_i, y_i, z_i)	第 i 个基站的位置
t	测量值
ω	测量噪声
$t_{i,LOS}$	手持终端与第 i 个基站间信号的视距传播时间
$E(x, y, z, t)$	误差函数
$A = \{m_1, m_2, \dots, m_n\}$	数据点集
Q	TOA 的协方差矩阵
α_i	加权因子

二、问题分析

建立不同环境下的无线通信基站的 TOA 的测量值与真实值之间的关系是基于无线通信进行终端定位的需要解决的首要问题。其次，通过

3.1 对问题 1 的分析

问题 1 中提供了 10 组在 LOS 或 NLOS 传播环境下从手持终端到基站的 TOA 测

量数据和所有基站的三维坐标，并要求根据这些测量出的数据，采用数学建模特别是测量模型和误差分析等方法，对所提供的数据样本加以分析，给出终端定位的模型。再次基础上，计算出终端的三维坐标，最终得以确定终端位置。本问题的难点主要有：

(1) 由题目中的背景可知，想要得到 LOS 或 NLOS 传播环境下的终端定位模型，需要引入对 TOA 的估计问题，现假设测量值为 t ，真实值为 \hat{t} ，当无线电波沿直线传播时测量值与真实值之间的关系为 $t = \hat{t} + \omega$ ，其中 ω 表示测量噪声；而在传播环境比较复杂时，观测值表示为 $t = f(\hat{t}, \omega)$ ，如何根据测量值 t 在某个恰当的意义下估计真实值 \hat{t} 的问题是一个难点。

(2) 通信基站的目标区域往往是高楼林立的城区，建筑物内部以及地下停车场等区域，这就导致场景基站、终端密集。如何基于 TOA 进行几何定位终端是本问的另一个难点。

通过以上问题分析可知，解决问题的思路：一是选取合适的方法，一般而言场景不同，函数 f 不同，思想有可能不同。从物理意义上看，对 \hat{t} 的估计相对式来说会变得困难，但数值解很容易得到，函数优化是整个建模的关键；二是基于 TOA 的几何定位我们建立非线性方程组，是建模过程中与场景匹配模型预算法。

3.2 对问题 2 的分析

问题 2 中提供了 10 组从手持终端到基站的 TOA 测量数据和所有基站的三维坐标，要求在近似最优三维定位精度的情况下尽可能的减少终端定位所需基站数目。根据给定的 10 组数据，基于终端定位模型 1，可以定位出终端的三维坐标。在确保定位精度的前提下，问题的目标是实现基站数目的最小化。因此，需要对定位终端的基站进行聚类分析，从而找出最优化模型。

3.3 对问题 3 的分析

问题 3 中提供了 5 组终端运动过程中采集到的 TOA 数据和所有基站的三维坐标，要求计算出终端的运动轨迹。问题 1 和问题 2 都是考虑终端在某一时刻的定位，而问题 3 是通过终端移动过程的某些时刻的定位来预测终端在某一时间段的运动轨迹。难点是如何去除一些孤立点去近似逼近终端的真实运动轨迹。该问题是求解二维平面上物体的运动轨迹。本文主要通过三次多项式进行对原始轨迹曲线进行平滑压缩处理。

3.4 对问题 4 的分析

问题 4 中提供了 5 组测量数据集，首先要求设计算法找到可被基站定位的所有终端。实际生活中，基站通信覆盖范围是有限的，很有可能造成给定区域内终端到某些基站的距离不可知，这将使得理论上某些终端可以被一定数目的基站定位，但实际上这些基站中可能存在一些基站与终端的距离已经超过基站的通信覆盖范围，最终使得终端无法测量。为了避免此类问题，同时高效利用通信资源，问题 4 进一步要求建立终端的平均连接度数 λ 与定位精度之间的关

系的数学模型。

四、问题 1 模型的建立与求解

基于网络的无线定位方法主要有基于信号到达强度（SOA）、信号到达时间（TOA）、信号到达角度（AOA）的定位算法。本文主要是基于 TOA 定位终端坐标。首先，建立不同场景下的 TOA 测量值与视距情况下的真实值的关系是解决基于无线基站通信终端定位的首要问题。其次，通过 TOA 参数的几何定位方法来确定终端的位置。

4.1 TOA 时间测量基本模型

在实际场景中，因为时钟同步以及多径传播等因素对 TOA 测量值有一定的影响，而且随着距离的增大，TOA 延时呈线性增加。

设手持终端到第 i 个基站的信号波达时间为

$$t_{i,m} = t_{i,LOS} + t_{i,e} + t_{i,N} \quad i=1, 2, \dots, M \quad (1)$$

式中， $t_{i,LOS}$ 是手持终端与第 i 个基站间信号的视距传播时间； $t_{i,e}$ 是系统测量误差，一般服从 0 均值的高斯分布； $t_{i,N}$ 是由 NLOS 传播等引入的误差，可服从指数分布、均匀分布以及 DELTA 分布的随机变量表示，本文仅考虑 $t_{i,N}$ 服从指数分布的情形，即

$$p(t_{i,N} / \tau_{i,ms}) = \begin{cases} \frac{1}{\tau_{i,ms}} \exp\left(-\frac{t_{i,N}}{\tau_{i,ms}}\right) & t_{i,N} \geq 0 \\ 0 & \text{其他} \end{cases} \quad (2)$$

式中， $\tau_{i,ms} = t_1 d_i^\varepsilon \xi$ 为传播时延扩展的均方根值； t_1 为 $\tau_{i,ms}$ 在 $d_i = 1km$ 时的中值； d_i 为收发间的距离；指数 $\varepsilon = 0.5 \sim 1$ ；随机变量 ξ 服从对数正态分布^[1]。

由概率论可知， $t_{i,N}$ 的概率密度函数为

$$\begin{aligned} p(t_{i,N}) &= \int_{-\infty}^{\infty} p(t_{i,N}, \tau_{i,ms}) d\tau_{i,ms} \\ &= \begin{cases} \int_0^{\infty} \frac{1}{\sqrt{2\pi}\sigma_z \tau_{i,ms}^2} \exp\left(-\frac{t_{i,N}}{\tau_{i,ms}} - \frac{(\ln \tau_{i,ms} - m_z)^2}{2\sigma_z^2}\right) d\tau_{i,ms} & t_{i,N} \geq 0 \\ 0, & t_{i,N} < 0 \end{cases} \quad (3) \end{aligned}$$

式中， $m_z = \ln t_1 + \varepsilon \ln d_i$ ， $\sigma_z^2 = \sigma_\xi^2$ 。容易得到 $t_{i,N}$ 的各阶统计量：

$$E(t_{i,N}^n) = n! t_1^n d_i^{n\varepsilon} e^{n^2 \sigma_z^2 \xi / 2} \quad n=1, 2, \dots \quad (4)$$

从式（4）可以看出，NLOS 传播时延扩的各阶统计量仅仅是通信距离的指数函

数。因此本文在 LOS 或 NLOS 环境下，延时等于观测 TOA 值 \hat{t} 减去视距 TOA 值 t ，通过模拟 sample_case001 中延时与观测 TOA 值 \hat{t} 发现成线性关系。实验模拟结果见图 2。从图 2 中可以看出测量时间与延时几乎成线性关系，即延时随着测量时间的增大而增大，其他样本的实验效果与该图基本类似。

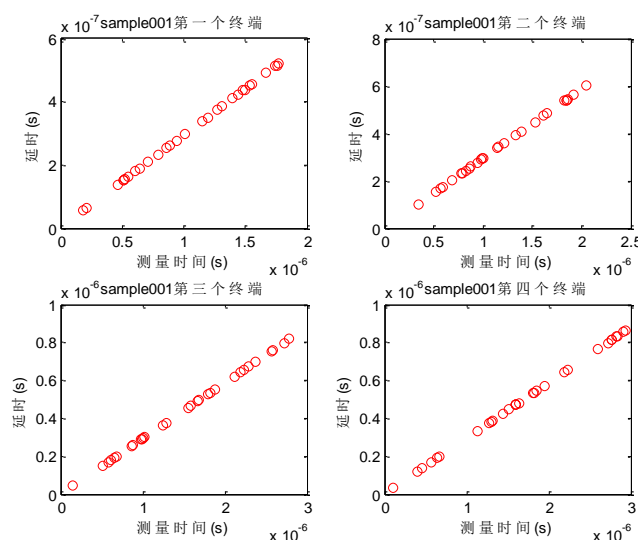


图 2 sample_case001 测量时间与延时关系图

根据图 2 结果，发现延时与 TOA 的测量值存在线性关系。文献[3,4]也研究发现延时值可用 TOA 的观测值线性表示。因此，我们建立 TOA 延时 \hat{t} 与观测值 t 之间的一次线性表达式，即 $\hat{t} = at + b$ ，其中 a, b 是与场景有关的常数。

4.2 三维 TOA 定位算法

本文研究三维情况下室内终端定位问题。目前，从事室内定位和导航服务的方法存在一定的局限性。因此，我们首先分析平面终端定位情形，再建立三维定位模型。

在二维平面内，终端精确定位需要 3 个基站。但实际情况一般存在测量误差，3 个基站可能有唯一解、可能无解、也可能多解。具体情况如图 3：

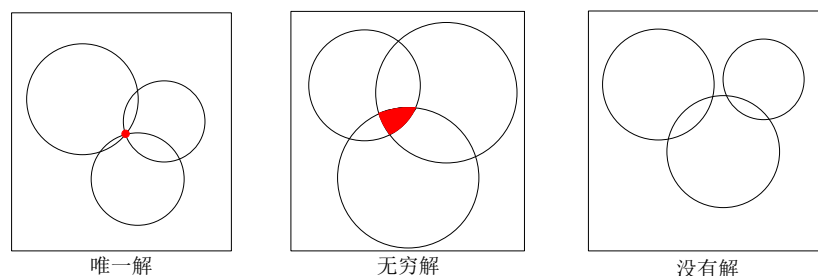


图 3 二维终端定位图解

但三维情况下精确定位终端至少需要 4 个基站。考虑到问题 1 中所测量的 TOA 数据值比视距环境下 TOA 的值大，因此用上述方法定位时必须满足下列非线性方程组：

$$\begin{cases} (x-x_1)^2 + (y-y_1)^2 + (z-z_1)^2 \leq r_1^2 \\ (x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2 \leq r_i^2 \\ \dots\dots\dots \\ (x-x_m)^2 + (y-y_m)^2 + (z-z_m)^2 \leq r_m^2 \end{cases}$$

其中， (x_i, y_i, z_i) 表示第*i*个基站的坐标， (x, y, z) 表示待定位终端的坐标。

采用TOA技术对手持终端进行定位时，只要测得TOA值，就可以得到手持终端到基站之间的距离，多个TOA测量值就可以在三维坐标中构成一组关于终端位置的圆方程^[2]。假定在三维空间中任意的分布着M个终端，待定位的终端坐标为 (x, y, z) 第*i*个BS的位置为 (X_i, Y_i, Z_i) ，则根据测量得到的TOA测量值 τ_i 可建立方程

$$\begin{aligned} R_i^2 &= (X_i - x)^2 + (Y_i - y)^2 + (Z_i - z)^2 \\ &= K_i - 2X_i x - 2Y_i y - 2Z_i z + R = (c\tau)^2 \quad i=1, 2, \dots, M \end{aligned} \quad (5)$$

其中： $K_i = X_i^2 + Y_i^2 + Z_i^2, i=1, 2, 3, \dots, R = x^2 + y^2 + z^2, c$ —电波传播速度，M 是TOA测量值数目。

令 $z_a = [z_p^T, R]^T$ 为未知适量，其中 $z_p = [x, y, z]^T$ ，从式（5）可建立以 z_a 为变量的线性方程组

$$h = G_a z_a \quad (6)$$

则MS估计位置对应的误差矢量为

$$\psi = h - G_a z_a^0 \quad (7)$$

式中： z_a^0 表示MS实际位置对应的 z_a 值

$$h = \begin{bmatrix} R_1^2 - K_1 \\ R_2^2 - K_2 \\ \vdots \\ R_M^2 - K_M \end{bmatrix}, G_a = \begin{bmatrix} -2X_1 & -2Y_1 & -2Z_1 & 1 \\ -2X_2 & -2Y_2 & -2Z_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ -2X_M & -2Y_M & -2Z_M & 1 \end{bmatrix}$$

采用加权平均最小二乘法（WLS），用TOA测量值的协方差矩阵Q近似替代误差 ψ 的协方差矩阵，可得

$$z_a = \arg \min \{ (h - G_a z_a)^T Q^{-1} (h - G_a z_a) \} = (G_a^T Q^{-1} G_a)^{-1} (G_a^T Q^{-1} h) \quad (8)$$

z_a 中 (x, y, z) 即为MS的近似估计位置。假定各TOA测量值相互独立，则（8）中Q矩阵为一对角矩阵

$$Q = \text{diag}\{\sigma_1^2, \sigma_2^2, \dots, \sigma_M^2\} \quad (9)$$

在误差方程（7）中，由于 z_a 中的 R 实际上是与 (x, y, z) 有关的量，用 Q 矩阵近似替代误差矢量 ψ 的协方差矩阵也会带来一定误差。为了得到更精确的估计位置，可采用与 *Chan* 算法类似的处理方法：对于方程(7)，当 *TOA* 误差较小时， M 个 *TOA* 测量值对应的误差矢量为

$$\begin{aligned}\psi &= 2Bn + n \odot n \approx 2Bn \\ B &= \text{diag}\{R_1^0, R_2^0, \dots, R_M^0\}\end{aligned}\quad (10)$$

其中： R_i^0 表示MS和第 i 个BS的实际距离， n 为以距离为单位的TOA测量误差（近似服从正态分布）。

式（7）中由TOA测量值构造的误差矢量 ψ 的协方差矩阵

$$\Psi = E[\psi\psi^T] = 4BQB \quad (11)$$

其中： Q 表示 *TOA* 的协方差矩阵。

为了得到B矩阵，可将测量得到的 R_i 代替 R_i^0 ，则 z_a 的第一次WLS估计值为

$$z_a = (G_a^T \Psi^{-1} G_a)^{-1} G_a^T \Psi^{-1} h \quad (12)$$

利用 z_a 值就可以得到新的 B 矩阵。利用以上过程再进行一次 *WLS* 计算就能得到改进的估计位置。以上计算假定 z_a 的各元素间相互独立，但事实上 R 与 *MS* 位置 (x, y, z) 有关，我们可利用这一关系得到更准确的位置估计。首先计算估计位置 z_a 的协方差矩阵，在有噪声情况下

$$R_i = R_i^0 + cn_i, G_a = G_a^0 + \nabla G_a, h = h^0 + \nabla h \quad (13)$$

由于 $G_a^0 z_a^0 = h^0$ ，式（6）表明

$$\psi = \Delta h - \Delta G_a z_a^0 \quad (14)$$

令 $z_a = z_a^0 + \Delta z_a$ ，再利用式（10）和（14）， Δz_a 及其协方差矩阵为

$$\begin{aligned}\Delta z_a &= c(G_a^T \Psi^{-1} G_a)^{-1} G_a^T \Psi^{-1} h B_n \\ \text{cov}(\Delta z_a) &= E[\Delta z_a, \Delta z_a^T] = (G_a^T \Psi^{-1} G_a)^{-1}\end{aligned}\quad (15)$$

矢量 z_a 为一均值为实际值，协方差矩阵由（15）确定的随机变量，因此 z_a 的元

素可表示为

$$z_{a,1} = x^0 + e_1, z_{a,2} = x^0 + e_2, z_{a,3} = x^0 + e_3, z_{a,4} = x^0 + e_4 \quad (16)$$

其中： e_1, e_2, e_3, e_4 表示 z_a 的估计误差， z_a 的前3个元素减去 X_1, Y_1, Z_1 , 再对各元素求平方。根据 (x, y, z) 和 R 的相互关系 $R = x^2 + y^2 + z^2$ ，可构造新的误差矢量 ψ' 为

$$\psi' = h' - G_a' z_a' \quad (17)$$

$$\text{式中： } h' = \begin{bmatrix} Z_{a,1}^2 \\ Z_{a,2}^2 \\ Z_{a,3}^2 \\ Z_{a,4}^2 \end{bmatrix}, \quad G_a' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad z_a' = \begin{bmatrix} x^2 \\ y^2 \\ z^2 \end{bmatrix}。$$

此处 ψ' 定义为 z_a 的估计误差，将式 (16) 带入式 (7) 中得

$$\begin{aligned} \psi'_1 &= 2x^0 e_1 + e_1^2 \approx 2x^0 e_1; \quad \psi'_2 = 2y^0 e_2 + e_2^2 \approx 2y^0 e_2 \\ \psi'_3 &= 2z^0 e_3 + e_3^2 \approx 2z^0 e_3; \quad \psi'_4 = e_4 \end{aligned} \quad (18)$$

误差的协方差矩阵为

$$\Psi' = E[\psi' \psi'^T] = 4B' \text{cov}(z_a) B'; \quad B' = \text{diag}\{x^0, y^0, z^0, \frac{1}{2}\} \quad (19)$$

B' 中 (x'', y'', z'') 可由 z_a 中的值近似代替， z_a' 的 WLS 估计为

$$z_a' = (G_a' \Psi'^{-1} G_a')^{-1} (G_a'^T \Psi'^{-1} h') \quad (20)$$

最终 MS 的定位计算结果为

$$z_p = \sqrt{z_a'} \text{ 或者 } z_p = -\sqrt{z_a'} \quad (21)$$

式中： z_p 中 (x, y, z) 正负号的选取应分别与 (12) 式 z_a 中对应元素一致，由此可消除定位结果的模糊性。

4.3 误差模型

基于 NL-LS 方法是利用多个 BS 的 TOA 参数及几何位置建立误差函数，然后利用泰勒级数展开方法进行近似计算使得误差函数最小，由此确定终端的几何位置。

误差模型为

$$E(x, y, z, t) = \sum_{i=1}^m \alpha_i [E(x, y, z, t)] \quad (22)$$

其中 $E_i(x, y, z, t_i) = ct_i - \sqrt{(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2}$ 表示第 i 个 BS 的误差函数；

m 为确定 MS 位置所采用 BS 的数量； α_i 表示加权因子，为第 i 个 BS 接收 MS 定位信号的强度^[3,4]。

如果基站的地理位置测量偏差较小，TOA 参数估计偏差也较小时，误差函数趋近于零。但是实际测量中由于受带宽、信噪比、时钟同步以及 NLOS 传播环境的影响，因此 TOA 参数估计偏差存在是必然的，此时 (22) 式的误差函数由最小值时，将会得到 MS 位置的最优解。因此，终端定位坐标可以通过公式 (23) 求得。

$$\min E(x, y, z, t) = 0 \quad (23)$$

为了便于求解误差函数的最小值，即寻求更好的初值点，现利用泰勒级数展开定理，式 (22) 整理为

$$\begin{aligned} E(x, y, z, t) \approx & \sum_{i=1}^m W_i \{ E_i(x_0, y_0, z_0, t_0) + \frac{(x_i - x)(x - x_0)}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2}} + \\ & \frac{(y_i - y)(y - y_0)}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2}} + \frac{(z_i - z)(z - z_0)}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2}} \\ & - c(t - t_0) \}^2 \end{aligned} \quad (24)$$

其中 (x_0, y_0, z_0, t_0) 接近 $E(x, y, z, t)$ 极值点处的参考点。此时 $E(x, y, z, t)$ 是 (x, y, z, t) 的二次函数。令

$$\frac{\partial E(x, y, z, t)}{\partial x} = 0 \quad (25)$$

$$\frac{\partial E(x, y, z, t)}{\partial y} = 0 \quad (26)$$

$$\frac{\partial E(x, y, z, t)}{\partial z} = 0 \quad (27)$$

$$\frac{\partial E(x, y, z, t)}{\partial t} = 0 \quad (28)$$

当方程组 (25) 到 (28) 相互独立时，求解方程组可得 $E(x, y, z, t)$ 的极值点 (x_m, y_m, z_m, t_m) 。将 (x_m, y_m, z_m, t_m) 作为新的参考点重新对方程组求解，再将新的解作为下次计算的参考点。这种迭代计算过程直至 $|\Delta x| + |\Delta y| + |\Delta z| + |\Delta t|$ 小于预先规定的最小值时结束。 (x, y, z, t) 的最后一次求解用来确定 MS 的位置。同时，为了减少终端定位的计算时间，本文利用网格破分法进行区域划分，基于贪心算法进行搜索，搜索过程如图 4 所示：

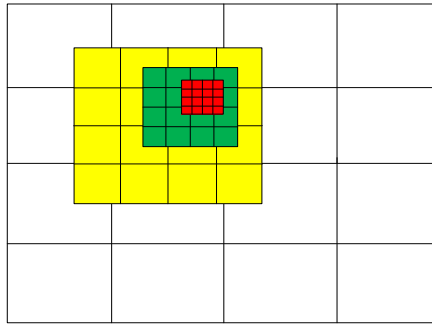


图 4 终端初始值定位搜索示意图

在网格破分后的区域，贪心算法可以较迅速地定位到初始坐标。

4.4 实验结果与分析

在不同场景下，受带宽、信噪比、时钟同步以及NLOS传播环境的影响，TOA测量值与视距传播环境下的真实传播时间存在差距。基于LOS或NLOS 传播环境分析TOA测量值与真实值之间的关系，建立了时间关系函数，进而通过几何三维定位方法建立非线性方程组求解终端的位置坐标。

根据问题1中case001-case010文件所给数据，用本文所提最小二乘方法进行终端定位。编号case001-case010文件提供TOA数据的基站情况为30，40，50，60，20，30，40，50，60，20；终端情况为1100，1200，1300，1400，1000，1100，1200，1300，1400，1000。实验环境为：Windows 系统和Mat lab 2012a，CPU为i3-4170M 和内存4GB。利用 Matlab 进行编程验证算法的有效性，程序见附录1。我们选取case001、case005 和case010中的部分终端信息进行展示，分别见表1-3。

表 1 case001 部分样本坐标斜率误差信息表

终端编号	X 轴(m)	Y 轴(m)	Z 轴(m)	斜率	误差
1	-284.68	89.72465	3.825084	0.264595	1.164487
2	100.2032	60.60742	2.98798	0.264494	1.184375
3	-45.758	-4.04079	0.489707	0.264398	0.734961
4	11.47692	157.9577	4.38342	0.264637	1.265502
5	-18.6396	294.3606	6.787967	0.264506	1.135767
6	-48.0652	-35.1832	1.613182	0.264391	1.445675
7	-312.863	276.6708	6.983941	0.26477	1.496503
8	146.0239	-274.352	2.174217	0.264867	1.102088
9	-200.892	-320.003	7.933903	0.264438	1.390618
10	105.3581	-85.4845	5.437536	0.264597	1.308651

表 2 case005 部分样本坐标斜率误差信息表

终端编号	X 轴(m)	Y 轴(m)	Z 轴(m)	斜率	误差
1	228.6823	-189.629	0.207545	0.046102	1.499074
2	184.477	-271.69	1.185685	0.045975	1.317168
3	180.0946	-263.835	4.130938	0.046211	0.910941
4	-421.568	-355.502	-8.9036	0.046374	1.86189
5	57.29563	22.68895	-0.15766	0.04614	1.4531
6	-196.269	65.3149	-1.89885	0.045967	1.304825

7	-247.709	-346.275	-3.85523	0.046324	1.11246
8	-228.088	399.7579	6.334754	0.045805	0.880986
9	-434.509	266.6839	6.004799	0.046219	1.142851
10	-323.664	-79.2222	-3.43109	0.046044	0.882526

表 3 case010 部分样本坐标斜率误差信息表

终端编号	X 轴(m)	Y 轴(m)	Z 轴(m)	斜率	误差
1	272.5268	232.3975	3.226032	0.005559	1.57454
2	-20.6886	232.9373	1.422231	0.006488	1.540445
3	-265.393	74.05819	-0.42114	0.005585	1.085439
4	12.42205	-40.3421	0.39248	0.005984	0.955902
5	-153.429	-131.68	-3.34059	0.005566	1.933656
6	-121.63	-180.902	3.081964	0.006204	1.508568
7	237.4713	253.3106	5.969599	0.006316	1.237902
8	122.3465	61.50116	-1.03306	0.00569	1.401463
9	193.6101	-202.575	9.162965	0.005813	2.192938
10	-26.1278	121.3037	-0.04483	0.005851	1.738322

下面用sample_case001- sample_case005验证算法的有效性，由于篇幅有限，现将sample_case001中的部分数据比较结果见表4，其他结果见附件1。

表 4 sample_case001 部分三维坐标计算值与实际值比较（米）

终端	实际值			计算值			识别精度
	X	Y	Z	X	Y	Z	
1	-21.19	4.48	1.48	-21.2535	4.525023	-0.55347	2.034959
2	-81.14	58.24	1.38	-81.1347	58.30078	1.208476	0.182049
3	-96.13	-215.22	1.02	-96.0355	-215.139	2.983624	1.967557
4	-296.06	-20.15	1.59	-296.064	-20.1589	3.732542	2.142564
5	86.29	-111.06	1.66	86.24028	-111.038	4.097538	2.438143
6	-31.26	244.68	1.30	-31.2335	244.6613	3.345796	2.046053
7	-286.47	-38.75	1.73	-286.524	-38.7939	2.807815	1.08008
8	319.30	-170.11	1.03	319.2005	-170.134	5.443885	4.41507
9	-37.40	273.59	1.73	-37.4817	273.6655	4.230713	2.503183
10	255.34	66.63	1.21	255.3354	66.6359	2.978548	1.768564

从表1中可以看出，sample_case001样本数据大部分的精度在3米范围以内，只有第八个终端的精度超过了3米，误差值也在0.5以内，而且终端的坐标x, y值的误差很小，主要误差出现在z值上。

case001- case004的实验结果散点图如图5-8.具体结果见附件1。

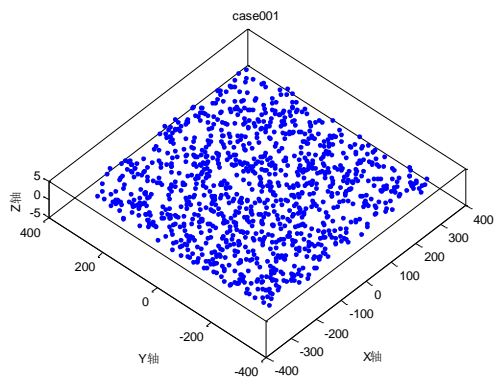


图 5 case001 的散点图

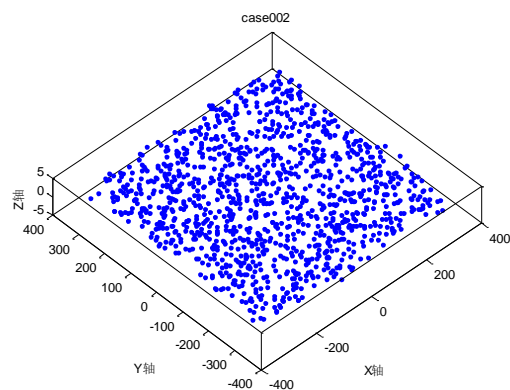


图 6 case002 的散点图

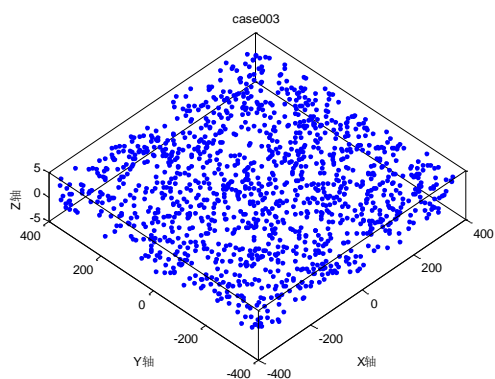


图 7 case003 的散点图

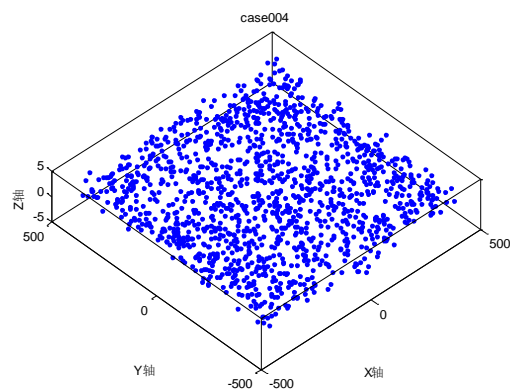


图 8 case004 的散点图

从图 5-8 可以看出，这四种场景的移动终端分布比较分散，没有太过密集的区域，所有终端在 X 轴和 Y 轴的分布基本一样，case001-case010 的数据结果见附件。

考虑到移动终端太密集，将所有终端的真实位置与计算位置展现在一张图片上把，不便于读者观察，所以现将sample_case001到sample_case005的部分终端的三维坐标以图形展示出，即图9-13所示：

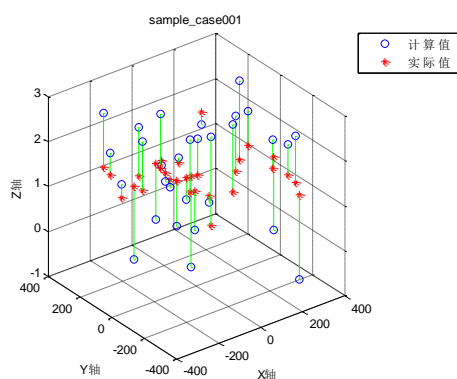


图 9 sample_case001 的部分终端的三维坐标对比

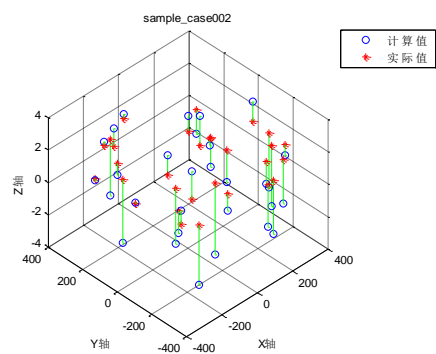


图 10 sample_case002 的部分终端的三维坐标对比

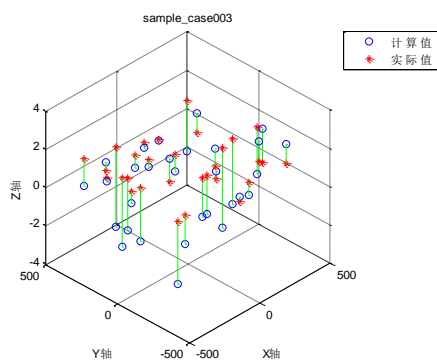


图 11 sample_case005 的部分终端的三维坐标对比

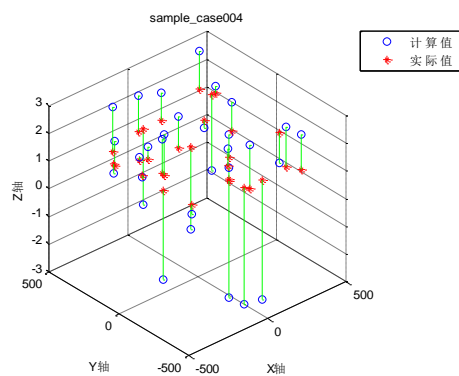


图 12 sample_case004 的部分终端的三维坐标对比

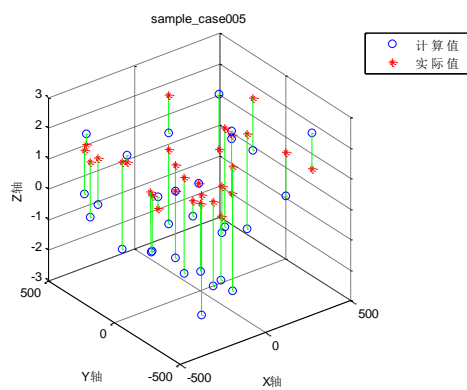


图13 sample_case005的部分终端的三维坐标对比

从图 9-13 可以看出，用我们所提出的定位方法求出的终端坐标与 sample_case001_ans-sample_case005_ans 文件给出的终端实际坐标相比，横纵坐标几乎一样，Z-轴坐标略有偏差，但误差基本都在 3 米以内。

五、问题 2 模型的建立与求解

5.1 终端定位优化算法分析与设计

在没有给出移动终端的位置时，我们可以近似认为运用全部基站数进行定位的坐标作为近似最优的参考标准，设某个移动终端运用全部终端定位的位置为 $A=(x,y,z)$ ，在减少 k 个基站后这个终端的定位位置为 $A_k=(x_k,y_k,z_k)$ ，定义

近似定位偏移 $LM = \sqrt{(x-x_k)^2 + (y-y_k)^2 + (z-z_k)^2}$ ，如果 $LM=0$ 或者比较小，

我们可以认为该位置终端用全部基站或部分定位基站定位的效果基本一样。如果整网络中近似定位偏移 LM 都很小，我们可以认为该网络中的所有终端定位都可以用部分基站而不用全部基站，平均近似定位偏移 ALM 为该网络中的全部 LM 的平均值。特别的当我们一个一个减少基站时，我们可以用 ALM 的突变判断某一个基站是否可以删除。一般的当 ALM 太大或者首次发生突然跳跃就不在减少基站数目。

在给出产考点的网络中，我们可以用实际坐标点与求解坐标的距离来判断求解坐标点的是否满足人们的要求。整个网络中所有坐标的实际坐标点与求解坐标点的距离的平均值来判断该求解结果是否总体满足精度要求，记为绝对距离平均值 ARD 。

通过模型 1，实现了无线通信基站定位终端的目标。如何用尽可能少的基站来实现终端的定位是现代商用通信网络对于三维定位的需求。本文通过基于最小生成树的 K-means 算法来实现终端定位的基站最优化^[5]。该算法利用数据的分布动态选取初始聚类中心，找出数据对象中分布比较密集的区域，使得初始的聚类中心接近结果聚类簇的中心，加速聚类过程。减少聚类的迭代次数，提高效率，并能得到稳定的聚类结果，取得较高的分类准确率。

聚类方法是通过最小生成树算法首先排除集合中的孤立点和噪音数据，减少这些数据对聚类过程的影响。然后利用样本集合的距离矩阵构造最小生成树，最后是对树进行剪枝，划分成 k 棵子树，每棵子树代表一个初始聚类簇，而子树中数据的算数均值作为初始的聚类中心。将 TD 算法得到的 k 个数据中心作为 K-means 算法的初始聚类中心，法的基本思想是：首先从 n 个样本集中随机选择 k 个样本作为初始聚类中心，根据每个样本与聚类中心的相似度，将其分配给最相似的聚类中心，得到 k 个互不相交的簇集合；然后重新计算每个簇的新中心，再将每个样本根据相似性原理分配给最近的簇中心重新计算每个簇的新中心，分配每个样本到距离最近的类簇。这个过程不断重复，直到各个簇的中心不再变化，得到原始样本集合的 k 个互不相交的稳定的类簇。该算法理论可靠、算法简单、收敛速度快并且能有效处理大量数据集。

对基站聚类后，选取尽可能少的基站定位终端的算法设计如下：

算法 1：终端定位的基站优化算法

输入：case011 到 case-020 的数据

输出：选出的 K 个基站

Step1 基于导入的基站坐标，对基站进行聚 K 类。

Step2 选出每一类中离中心最近的基站作为代表。

Step3 判定基站是否满足终端定位的要求。

Step4 是，对选出的 K 个基站重新聚成 $K-1$ 类，计算 $K-1$ 个基站的精度；
否则，扩大 K ，返回 Step1 重新聚类。

Step5 判断基站定位精度能否满足要求。

Step6 满足，把 $K-1$ 赋值给 K ，返回 Step4；否则，得到最小基站数 K 。

通过算法 1，我们知道终端定位的主要目的是在保证定位精度的前提下，减少基站的数目，提高基站的利用率，从而提高室内三维定位的更高价值。终端定位的基站优化算法的流程图如图 14 所示。

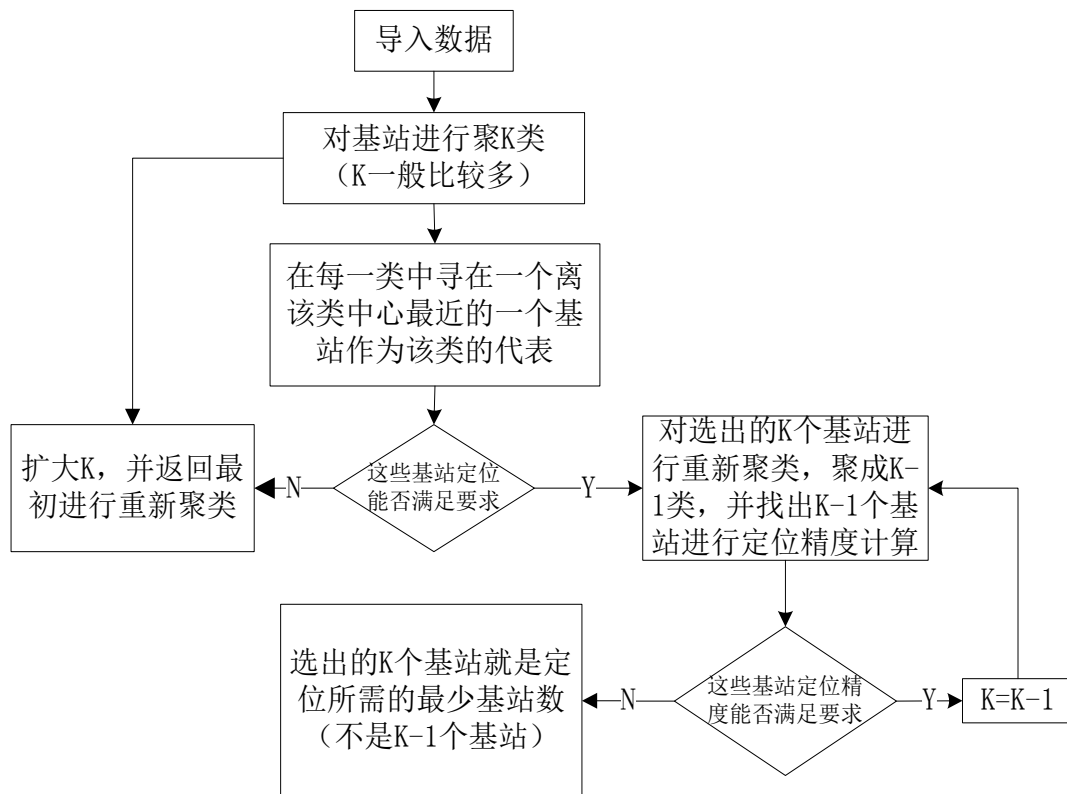


图 14 终端定位的基站优化算法流程图

5.2 实验结果与分析

编号 case011-case020 文件提供 TOA 数据的相关基站个数情况为 30, 40, 50, 60, 20, 30, 40, 50, 60, 20；终端个数情况为 1100, 1200, 1300, 1400, 1000, 1100, 1200, 1300, 1400, 1000。实验环境为：Windows 系统和 Mat lab 2012a，CPU 为 i3-4170M 和内存 4GB。case010-case020 的数据进行描述。我们首先将每一个 case 中的基站数进行聚类，这里我们都取 $k=15$ ，运用算法 1 逐一减少基站，并且计算每次减少后的 ALM ，然后判断最有基站数。通过计算 sample 中的案例来检验近似定位精度。利用 Matlab 进行编程验证算法

的有效性，程序见附录 2。平均近似定位偏移量 ALM 见表 5。

表 5 平均近似定位偏移 ALM

基站数目	9	10	11	12	13	14	15
case011	7.14	5.27	3.21	2.87	2.43	2.45	2.21
case012	6.97	5.61	3.54	2.97	2.83	2.42	2.12
case013	8.04	6.76	5.93	3.81	3.76	3.41	2.98
case014	7.79	6.41	5.24	3.05	3.01	2.85	2.43
case015	7.45	5.02	2.87	2.54	2.37	2.21	2.18
case016	6.56	4.97	2.92	2.86	2.39	2.17	1.99
case017	6.48	5.12	3.34	3.21	3.04	2.91	2.75
case018	6.79	6.23	5.31	2.74	2.52	2.41	2.03
case019	6.81	5.42	5.27	3.12	3.09	2.96	2.30
case020	7.04	5.57	2.95	2.67	2.41	2.32	2.03

从表 5 可以看出，case011、case012、case015、case016、case017 和 case020 的 ALM 首次出现大的拐点在 10 和 11 之间，可以考虑用 10 个基站去实现近似最优的三维定位。case013、case014、case018 和 case019 的 ALM 首次出现大的拐点在 11 和 12 之间，可以考虑用 11 个基站去实现近似最优的三维定位。为了验证我们的结论，我们统一取 11 台基站作为最少基站数，运用 sample 的数据进行绝对量化测试。sample 中所有 case 的绝对距离平均值 ARD 见表 6。

表 6 sample 中所有 case 的 ARD

case	sample_case001	sample_case002	sample_case003	sample_case004	sample_case005
ARD	5.83	4.87	5.54	5.71	5.19

从表 6 中，我们可以得出所有的 ARD 都小于 6，在问题 1 中，我们的模型平均识别精度在 3 米多。所以在实现近似最优的三维定位中，我们可以取 11 基站作为最少基站数。

六、问题 3 模型的建立与求解

通过模型 1 可以用基站来定位终端在某一时刻的位置。而实际生活中，终端可能在不同时刻，所处位置不同。因此需要根据不同时刻定位到的终端位置，建立终端在移动过程中的运动轨迹模型^[6,7]。根据不同基站在每个时刻测得的终端的 TOA 数据，通过终端定位模型（即模型 1），可以得到每个终端在不同时刻的位置。即第 i 个基站 $(MS)_i$ 在 t_j 时刻的位置坐标为 $(x_{ij}, y_{ij})(i=1,2,\dots,n)$ ，得到

终端的原始轨迹由于测量误差可能存在运动抖动，需要通过平滑处理去探寻运动规律。本文基于三次多项式对原始轨迹曲线进行平滑压缩处理^[8]。

采用传统的多项式曲线、B 样条、C 样条或者曲线拟合虽然会在轮廓误差取得较好的效果，但是终端在运动过程中的 TOA 数据处理量庞大而复杂，效率低下。因此本文提出了基于三次多项式曲线的平滑压缩算法，该算法从运动路径中寻找出连续微小线段区域，并在满足精度的要求下将其拟合成平滑的三次多项式运动曲线，从而达到提高效率和较少所需处理的数据量。

6.1 终端轨迹定位算法

单纯的终端定位与一个终端在不同时刻呈现出来的位置的定位算法有所不同，不同终端定位的时，其他终端与该终端的位置几乎没有什么必然的联系，然而同一终端在不同时刻的位置相互之间存在着一定的联系，相邻间隔很短的两个时间内终端的位置差异不会太大。我们将从轨迹位置一般不会发生突变的情况下进行轨迹定位。

算法 2：终端运动轨迹算法

输入：case021 到 case025 的数据

输出：终端的运动轨迹

Step1 利用第一问提出的终端定位模型求得终端在不同时刻的位置坐标。

Step2 选出跳跃较大的间断点，运用几个相邻的点的均值作为该点的新位置点。

Step3 重复 Step2 直到没有较大的间断点为止。

Step4 对于位置修改过大的点检验，判断它的位置是否某个基站 LOS 距离矛盾，如果矛盾则剔除该点，否则执行 Step5。

Step5 将剩下的跳跃较小的点进行三次多项式曲线回归拟合，然后将离群点拉回。

Step6 如果轨迹仍然有突变的地方，重复 Step4。

6.2 终端位置点拟合

假设某两个时刻终端的位置为 D_1, D_2 ，将其连接成一个连续微小线段，采用分段的三次多项式曲线进行拟合。现假设待拟合的连续微小线段运动区域中的数据点集为 $A = \{m_1, m_2, \dots, m_n\}$ ，从该点集 A 中，按间隔 k 选出多项式拟合所需的四个数据点，并将这四个数据点组成新的数据点集： $A = \{m_z, m_{z+k+1}, m_{z+2k+2}, m_{z+3k+3}\}$ ，其中 $k = 0, 1, \dots, n-3$ 。每一段三次多项式函数为：

$$f(p) = a + bp + cp^2 + dp^3 \quad (0 \leq p \leq pl) \quad (27)$$

令每段四个数据点分别对应 $p = 0, p = pl/3, p = 2pl/3, p = pl$ 时多项式的值，即可得到多项式的系数为：

$$[a, b, c, d] = [m_z, m_{z+k+1}, m_{z+2k+2}, m_{z+3k+3}] \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & \frac{pl}{3} & \frac{2pl}{3} & pl \\ 0 & (\frac{pl}{3})^2 & (\frac{2pl}{3})^2 & pl^2 \\ 0 & (\frac{pl}{3})^3 & (\frac{2pl}{3})^3 & pl^3 \end{bmatrix} \quad (28)$$

因此就可以求出该三次多项式。该多项式函数涵盖了集合 A 中

$m_z, m_{z+1}, \dots, m_{z+3k+3}$ 共 $3k+4$ 个数据点，这些点对应的参数值 p_i 和函数值 m'_i 为：

$$\begin{aligned} p_i &= \frac{pl \cdot (i-1)}{3 \cdot (k+1)} \quad i=1, 2, \dots, 3k+4 \\ m'_i &= f(p_i) \quad i=1, 2, \dots, 3k+4 \end{aligned} \quad (29)$$

在进行逼近精度误差控制时，可以以多项式的方差值是否小于方差的期望值 S 为条件进行判断；也可以以各个数据点对应多项式函数值 m'_i 的误差值是否小于期望误差 e 为条件进行判断，公式如下：

$$\sum_{i=1}^{3k+4} [m'_i - m_{z+i-1}]^2 < S \quad (30)$$

$$\max(|m_i - m_{z+i-1}|) < e \quad i=1, 2, \dots, 3k+4 \quad (31)$$

若多项式逼近精度符合要求，则将多项式的最后一个数据点作为下一段多项式的数据起点，即将第一个数据点的下标设置为 $z = z + m \cdot k + m$ ，并在数据点集 A 按照间隔点 k 选取下一段多项式逼近所需要的三个数据点，按照上述方法计算下一段多项式的系数。如果多项式的逼近精度大于 e ，则缩小间隔点 k 的值，使 $k = k - 1$ ，从数据点集 A 中选取所需要的三个数据点，再按照以上方法计算该多项式的系数，重复该过程直到多项式的逼近精度小于 e 。本文基于三次多项式对原始轨迹曲线进行平滑压缩处理简要过程如图 14 所示

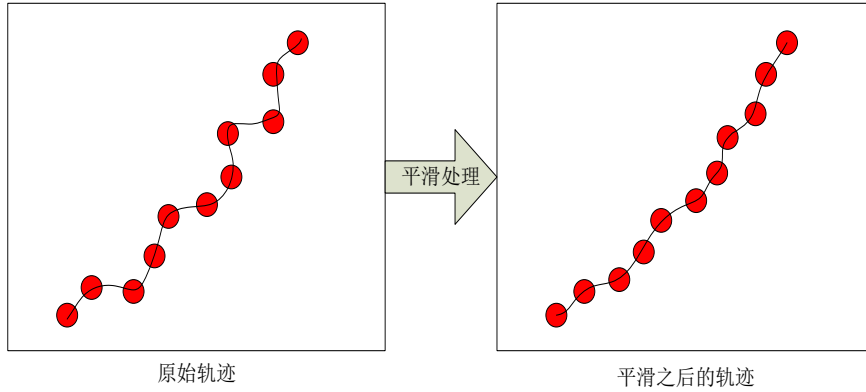


图 14 轨迹平滑处理图

6.3 实验结果与分析

编号 case021-case025 文件提供 TOA 数据的相关基站个数情况为 30, 40, 50, 60, 20, 30, 40, 50, 60, 20；终端个数情况为 1100, 1200, 1300, 1400, 1000, 1100, 1200, 1300, 1400, 1000。实验环境为：Windows 系统和 Mat lab 2012a，CPU 为 i3-4170M 和内存 4GB。如果一个时刻的位置在该位的 20 以内没有其他点，则认为该位置为跳跃较大的点。利用 Matlab 进行编程验证算法的有效性，程序见附录 3。case021 和 case022 部分终端的坐标见表 7。其他坐标见附件 1。

表 7 case021 和 case022 部分终端的坐标信息

case021 编号	X 轴	Y 轴	case022 编号	X 轴	Y 轴
1	-264.721	416.6347	1	-231.163	333.8204
2	-199.97	263.7675	2	-6.43307	7.757561
3	-152.689	173.5589	3	107.3646	-3.12367
4	-52.2313	41.00504	4	-155.781	179.5778
5	27.19763	-6.78074	5	-72.2826	61.43309
6	3.76667	1.998257	6	-124.481	128.7436
7	-150.944	170.4005	7	121.5289	2.800535
8	90.37189	-8.5786	8	-40.3148	31.19689
9	-118.008	118.6192	9	-145.118	161.389
10	28.65354	-7.19589	10	-298.001	508.6933

为了展示终端运动的轨迹情况，我们将 case021-case025 的运行结果展示在如图 15-19 所示。

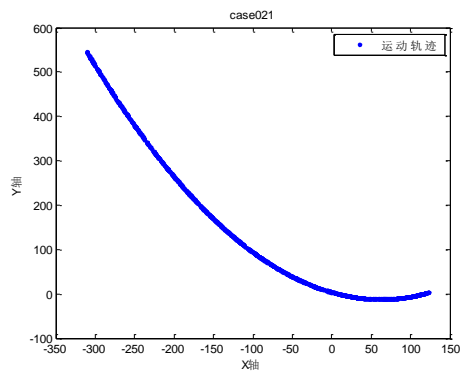


图 15 case021 的运动轨迹

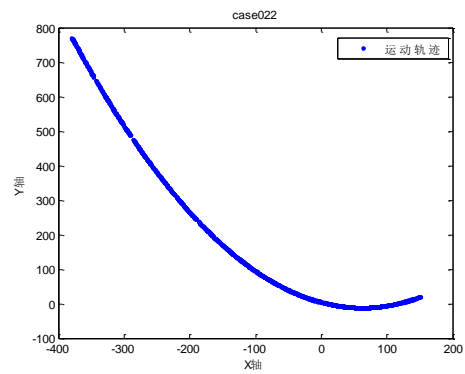


图 16 case022 的运动轨迹

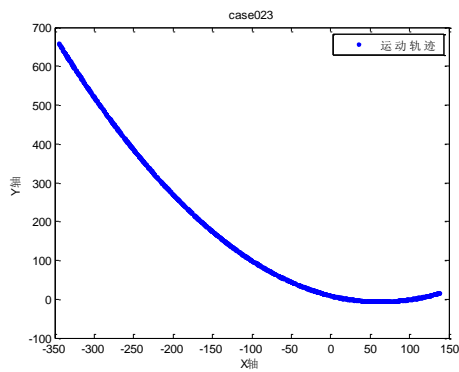


图 17 case023 的运动轨迹

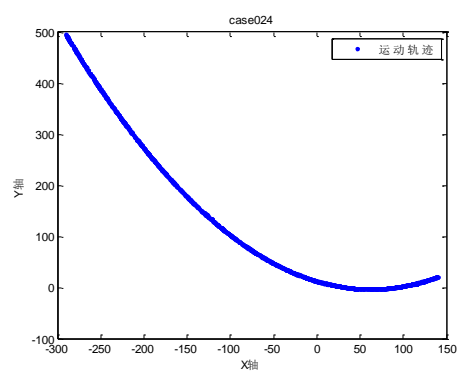


图 18 case024 的运动轨迹

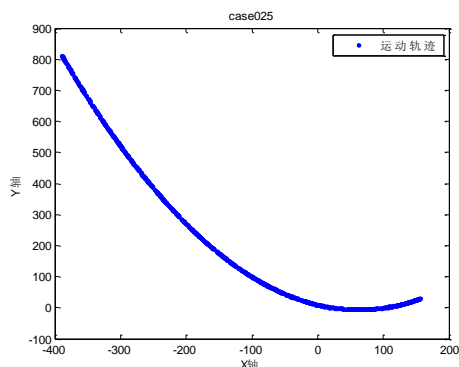


图 19 case025 的运动轨迹

从图 15-19 可以看出，这五种场景的移动终端的运动轨迹较平滑，只是 case022 的运动轨迹图中有两处缺省，有可能是数据出现了异常。通过对上面这些图的分析说明，我们设计的终端运动轨迹算法效果不错。

七、问题 4 模型的建立与求解

7.1 终端定位连接度与定位精度模型的建立

根据问题 2 所提出的方法，定位每个终端所需的最少基站数目可以得到。需要指出的是问题 1-3 假设给定区域内中终端到基站的距离都是可知的，这是一种理想状态的假设。在问题 1-3 中，对每个终端而言，其与所有基站之间都可以连接，只要满足一定的基站数目，该终端就一定可以被定为。但是，在实际中每个基站能过覆盖的范围是有限的，对于需要定位的终端而言，终端与基站的距离超过基站的覆盖范围，那么该基站与需要定位的终端之间无法连接。只有在通信半径覆盖范围内，一个终端实际所连接的基站数目不少于该终端定位所需要的基站数目，此基站才能被有效定位。

首先，根据问题二的模型和给定 case026-case030 的数据信息，可得定位在通信半径覆盖范围内，每个终端定位所需要的最少基站数目 s 。因此与该终端相连接的基站数目至少为应该 s 。另外根据每个终端和每个基站的坐标可知以该终端为中心，200 米为半径的范围内的基站的个数 h 。如果， $h \geq s$ ，那么该终端可被定位；否则，该终端无法定位。对于每个终端重复上述过程，可以找到此区域内所有可被定位的终端。从而得到可以被基站有效定位的所有终端。相应设计算法如下：

算法 3：寻找可以被基站定位的终端

输入：case026 到 case025 的数据

输出：可以被基站定位的终端数

Step1 通过问题 1 中的终端定位模型确定坐标（忽略 200 米通信半径）。

Step2 求解每个终端到所有基站的距离。

Step3 统计每个终端 200 米半径内的实际存在的基站数目 h ；

Step4 通过问题 2 中的模型确定每个终端在 200 米半径内所需要的最小基站数目 s 。

Step5 如果一个终端 200 米半径内的基站数目 h 大于等于定位该终端所需要的最少基站数目 s ，则认为该终端能被定位。否则不能被定位。

Step6 重复 Step3-5，找出能被定位的所有终端。

7.2 终端平均连接度数 λ 与定位精度关系模型的建立

首先，假设在整个网络区域中有个 m 终端， n 个基站。根据 7.1 中设计的算法，可以知道第 i 个终端的连接数目为 h_i ($i=1,2,\dots,m$)，所需的最少基站数目为 s_i 。对每个终端去判断其连接数目 h_i 是否大于所需基站数目 s_i 。如果 h_i 大于 s_i ，则计算该终端的连接度数 $\lambda_i = h_i$ ；否则该记终端连接度数为 0。最后求得终端平均连接度数 $\lambda = \sum_{i=1}^m \lambda_i / m$ 。其次，在以终端为中心，200 米为半径的覆盖范围内，当 $h_i < s_i$ 时，随着基站数目增加到 s_i ，定位精度会从 0 逐渐增大到一定值，然后不再随基站数目的增加而改变；当 $h_i \geq s_i$ 时，随着基站数目增加，定位精度不会发生改变。通过分析发现，终端平均连接度数 λ 随连接数的增大而减小；而定位精度随着 h_i 的增大先增加后不变。因此终端平均连接度数 λ 与定位精度 ψ 呈负相关。通过问题 1-4 的 case 文件和 sample 文件中所给的数据进行拟合观测，发现终端平均连接度数 λ 与定位精度 η 之间有如下关系

$$\eta = \alpha \arctan(\beta / \lambda) + \varepsilon \quad (34)$$

其中 α, β 为待求常数， ε 为随机参数。

7.3 结果与分析

编号 case026-case030 文件提供 TOA 数据的相关基站个数情况为 30, 40, 50, 60, 20；终端个数情况为 1100, 1200, 1300, 1400, 1000。实验环境为：Windows 系统和 Mat lab 2012a，CPU 为 i3-4170M 和内存 4GB。利用 matlab 编程进行求解，程序见附录 4。现将 case030 的部分能够被定位的终端的坐标展示在表 8，其他结果详见附件 1。

表 8 case030 的部分能够被定位的终端的坐标

终端编号	X 轴	Y 轴	Z 轴
9	99.038086	100.124437	2.542021
11	-218.019436	-125.883624	3.961487
12	229.384229	-109.458281	3.189406
29	-66.331824	-3.783027	-2.090686
48	103.572694	69.668045	1.629935
69	33.292792	137.299683	2.423697

case026-case030 的连接度数 λ 见表 9。

表 9 case026-case030 的连接度数 λ

case	case026	case027	case028	case029	case030
λ	0.1700	0.4917	0.5062	0.6400	0.2530

八、模型的评价与推广

8.1 模型评价

本文研究无线基站定位室内终端的问题。本论文的主要工作就是对某一时刻终端的定位和移动终端运动轨迹的追踪。针对问题 1，通过建立误差函数模型确定终端坐标，该模型的定位精度误差在 3 米左右，突破了 GPS 等商用卫星定位系统无法实现定位的场景，能够基于有噪音的测量得到位置信息的准确估计。同时，从侧面说明利用无线基站研究室内三维定位问题的方向是正确的、可行的。针对问题 2，设计终端定位的基站优化算法满足了现代商用通信网络用尽可能少的基站完成终端设备定位的需求，丰富了最优估计理论和优化算法等的研究，为室内室外融合定位导航系统的出现提供理论支持。针对问题 3，基于次多项式拟合的终端运动轨迹模型的建立，为移动终端的追踪技术的研究提供理论上的支持。

问题 4，对于定位精度和终端平均连接度关系模型的建立，有助于对基站资源的合理利用和网络布置的优化。另一方面，本文所建立的终端定位模型，终端定位的基站优化模型，终端运动轨迹模型和定位精度与连接度关系模型有待于通过更多的实践检验去改善模型的局限性。比如，以视距路径数位标准，所建立的终端定位模型适用于 NLOS 污染较轻（LOS 路径数 ≥ 3 ）的场景，但是在 NLOS 污染较重（LOS 路径数 < 3 ）的场景下该模型的定位精度可能会有所下降。因此，这些模型的实用性和定位精度都有待进一步研究。

8.2 模型推广

无线通信基站的三维定位既然能在高楼林立的城区，建筑物内部、地下停车场等区域应用，那么智能仓储、智能工厂、固定资产追踪等对于三维坐标信息敏感的垂直行业，以及传统运营商感兴趣的商场、办公楼中基于位置信息的室内导航、人群流量分析，基于精确三维地理位置信息的业务推送等服务提供基础性技术，通信基站将得到广泛的应用。高度智能化的通信基站也可能直接用于公安无线建设、管理、应急指挥和社会公众服务的三维地理信息平台，实现公安无线通信管理系统的信息化、模型化、数字化、可视化和科学化。

参考文献

- [1] 刘颖, 王树勋, 宋春莆. 移动终端定位算法及误差分析[J]. 系统工程与电子技术, 2001, 23(7):98-102.
- [2] 李招华, 汪毓铎, 邵青. 基于 Chan 的 TDOA 三维定位算法[J]. 现代电信科技, 2014(11):36-40.
- [3] Luoh L. ZigBee-based intelligent indoor positioning system soft computing[J]. Soft Computing, 2014, 18(3):443-456.
- [4] Moon G B, Chun S B, Hur M B, et al. A T-DMB navigation system for seamless positioning in both indoor and outdoor environments[J]. Eurasip Journal on

Advances in Signal Processing, 2014, 2014(1):1-16.

[5] 贾瑞玉, 李振. 基于最小生成树的层次 K-means 聚类算法[J]. 微电子学与计算机, 2016, 33(3):86-88.

[6] 潘迪夫, 刘辉, 李燕飞. 基于时间序列分析和卡尔曼滤波算法的风电场风速预测优化模型[J]. 电网技术, 2008, 32(7):82-86.

[7] 高海兵, 周驰, 高亮. 广义粒子群优化模型[J]. 计算机学报, 2005, 28(12): 1980-1987.

[8] 李浩, 黄艳, 马岩蔚. 基于三次多项式曲线的轨迹平滑压缩算法[J]. 组合机床与自动化加工技术, 2016(6).

附录

附录 1

```
function [f,g]= testf(x)
global path_location; %'..\data\sample_case002_location.txt'
c1 = importdata(path_location);
len1 = length(c1(:,1));
global ti;
global mk;
c = ti(mk,:);
k1=[0.029277507804620 1.543344122922481e-09];
for i=1:len1
    th(i)=(c(i)-(c(i)*x(4)+k1(2)))*3*10^8;
    th0(i)=c(i)*3*10^8; % %%%%%%%%%%
    y0(i)=(th(i)-sqrt((x(1)-c1(i,1))^2+(x(2)-c1(i,2))^2+(x(3)-c1(i,3))^2)
    )^2;

    g0(i)=2*(th(i)-sqrt((x(1)-c1(i,1))^2+(x(2)-c1(i,2))^2+(x(3)-c1(i,3))^2)
    )*(-2*
    (x(1)-c1(i,1))/sqrt((x(1)-c1(i,1))^2+(x(2)-c1(i,2))^2+(x(3)-c1(i,3))^2)
    );

    g1(i)=2*(th(i)-sqrt((x(1)-c1(i,1))^2+(x(2)-c1(i,2))^2+(x(3)-c1(i,3))^2)
    )*(-2*
    (x(2)-c1(i,2))/sqrt((x(1)-c1(i,1))^2+(x(2)-c1(i,2))^2+(x(3)-c1(i,3))^2)
    );

    g2(i)=2*(th(i)-sqrt((x(1)-c1(i,1))^2+(x(2)-c1(i,2))^2+(x(3)-c1(i,3))^2)
    )*(-2*
    (x(3)-c1(i,3))/sqrt((x(1)-c1(i,1))^2+(x(2)-c1(i,2))^2+(x(3)-c1(i,3))^2)
    );

    g4(i)=2*(th(i)-sqrt((x(1)-c1(i,1))^2+(x(2)-c1(i,2))^2+(x(3)-c1(i,3))^2)
    )*3*10^8*(-c(i));
end
f=sum(y0);
g=[sum(g0);sum(g1);sum(g2);sum(g4)];
%%%%%%%%%%%%%
function case001
clear all, close all %%%%%%%%%%kk
global path_location;
path_location = '..\data\case001_location.txt';
time = importdata('..\data\case001_toa.txt');
options = optimset('GradObj','on');
```

```

len = length(time(:,1));
global ti;
global mk;
ti = time;
weizhi = zeros(len,3);
for mk= 1:len
    mk
    [x,y]=fminunc('tesf',[-450 457 1 0.29],options);
    weizhi(mk,:) = x(1,1:3);
end
fid = fopen('..\results2\case001_output.txt','wt');
for i = 1: len
    fprintf(fid,'%f\t%f\t%f\n',weizhi(i,1:3));
end
fclose(fid);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function case002
clear all, close all %%%%%%%%%%kk
global path_location;
path_location = '..\data\case002_location.txt';
time = importdata('..\data\case002_toa.txt');
options = optimset('GradObj','on');
len = length(time(:,1));
global ti;
global mk;
ti = time;
weizhi = zeros(len,3);
for mk= 1:len
    mk
    [x,y]=fminunc('tesf',[-450 457 1 0.29],options);
    weizhi(mk,:) = x(1,1:3);
end
fid = fopen('..\results2\case002_output.txt','wt');
for i = 1: len
    fprintf(fid,'%f\t%f\t%f\n',weizhi(i,1:3));
end
fclose(fid);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function case003
clear all, close all %%%%%%%%%%kk
global path_location;
path_location = '..\data\case003_location.txt';

```

```

time = importdata('..\data\case003_toa.txt');
options = optimset('GradObj','on');
len = length(time(:,1));
global ti;
global mk;
ti = time;
weizhi = zeros(len,3);
for mk= 1:len
    mk
    [x,y]=fminunc('tesf',[-100 200 1 0.29],options);
    weizhi(mk,:) = x(1,1:3);
end
fid = fopen('..\results2\case003_output.txt','wt');
for i = 1: len
    fprintf(fid,'%f\t%f\t%f\n',weizhi(i,1:3));
end
fclose(fid);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function case004
clear all, close all %%%%%%%%%kk
global path_location;
path_location = '..\data\case004_location.txt';
time = importdata('..\data\case004_toa.txt');
options = optimset('GradObj','on');
len = length(time(:,1));
global ti;
global mk;
ti = time;
weizhi = zeros(len,3);
for mk= 1:len
    mk
    [x,y]=fminunc('tesf',[-450 457 1 0.29],options);
    weizhi(mk,:) = x(1,1:3);
end
fid = fopen('..\results2\case004_output.txt','wt');
for i = 1: len
    fprintf(fid,'%f\t%f\t%f\n',weizhi(i,1:3));
end
fclose(fid);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function case005
clear all, close all %%%%%%%%%kk

```

```

global path_location;
path_location = '..\data\case005_location.txt';
time = importdata('..\data\case005_toa.txt');
options = optimset('GradObj','on');
len = length(time(:,1));
global ti;
global mk;
ti = time;
weizhi = zeros(len,3);
for mk= 1:len
    mk
    [x,y]=fminunc('tesf',[-450 457 1 0.29],options);
    weizhi(mk,:) = x(1,1:3);
end
fid = fopen('..\results2\case005_output.txt','wt');
for i = 1: len
    fprintf(fid,'%f\t%f\t%f\n',weizhi(i,1:3));
end
fclose(fid);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function case006
clear all, close all %%%%%%%%%kk
global path_location;
path_location = '..\data\case006_location.txt';
time = importdata('..\data\case006_toa.txt');
options = optimset('GradObj','on');
len = length(time(:,1));
global ti;
global mk;
ti = time;
weizhi = zeros(len,3);
for mk= 1:len
    mk
    [x,y]=fminunc('tesf',[-450 457 1 0.29],options);
    weizhi(mk,:) = x(1,1:3);
end
fid = fopen('..\results2\case006_output.txt','wt');
for i = 1: len
    fprintf(fid,'%f\t%f\t%f\n',weizhi(i,1:3));
end
fclose(fid);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

function case007
clear all, close all %%%%%%%%%kk
global path_location;
path_location = '..\data\case007_location.txt';
time = importdata('..\data\case007_toa.txt');
options = optimset('GradObj','on');
len = length(time(:,1));
global ti;
global mk;
ti = time;
weizhi = zeros(len,3);
for mk= 1:len
    mk
    [x,y]=fminunc('tesf',[-450 457 1 0.29],options);
    weizhi(mk,:) = x(1,1:3);
end
fid = fopen('..\results2\case007_output.txt','wt');
for i = 1: len
    fprintf(fid,'%f\t%f\t%f\n',weizhi(i,1:3));
end
fclose(fid);
end
%%%%%%%%%%%%
function case008
clear all, close all %%%%%%%%%kk
global path_location;
path_location = '..\data\case008_location.txt';
time = importdata('..\data\case008_toa.txt');
options = optimset('GradObj','on');
len = length(time(:,1));
global ti;
global mk;
ti = time;
weizhi = zeros(len,3);
for mk= 1:len
    mk
    [x,y]=fminunc('tesf',[-450 457 1 0.29],options);
    weizhi(mk,:) = x(1,1:3);
end
fid = fopen('..\results2\case008_output.txt','wt');
for i = 1: len
    fprintf(fid,'%f\t%f\t%f\n',weizhi(i,1:3));
end
fclose(fid);

```

```

end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function case009
clear all, close all %%%%%%%%%%kk
global path_location;
path_location = '..\data\case009_location.txt';
time = importdata('..\data\case009_toa.txt');
options = optimset('GradObj','on');
len = length(time(:,1));
global ti;
global mk;
ti = time;
weizhi = zeros(len,3);
for mk= 1:len
    mk
    [x,y]=fminunc('tesf',[-450 457 1 0.29],options);
    weizhi(mk,:) = x(1,1:3);
end
fid = fopen('..\results2\case009_output.txt','wt');
for i = 1: len
    fprintf(fid,'%f\t%f\t%f\n',weizhi(i,1:3));
end
fclose(fid);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function case010
clear all, close all %%%%%%%%%%kk
global path_location;
path_location = '..\data\ case010_location.txt';
time = importdata('..\data\ case010_toa.txt');
options = optimset('GradObj','on');
len = length(time(:,1));
global ti;
global mk;
ti = time;
weizhi = zeros(len,3);
for mk= 1:len
    mk
    [x,y]=fminunc('tesf',[-200 200 1 0.29],options);
    weizhi(mk,:) = x(1,1:3);
end
fid = fopen('..\results2\case010_output.txt','wt');
for i = 1: len
    fprintf(fid,'%f\t%f\t%f\n',weizhi(i,1:3));

```



```

end
fclose(fid);
end
附录 2
%%%附录2中的函数都要用到附录1中的tesf函数
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function case011
clear all, close all %%%%%%%%%%kk
global path_location;
path_location = '..\data\case011_location.txt';
time = importdata('..\data\case011_toa.txt');
options = optimset('GradObj','on');
len = length(time(:,1));
global ti;
global mk;
ti = time;
weizhi = zeros(len,3);
for mk= 1:len
    mk
    [x,y]=fminunc('tesf',[-450 457 1 0.29],options);
    weizhi(mk,:) = x(1,1:3);
end
fid = fopen('..\results2\case011_output.txt','wt');
for i = 1: len
    fprintf(fid,'%f\t%f\t%f\n',weizhi(i,1:3));
end
fclose(fid);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function case012
clear all, close all %%%%%%%%%%kk
global path_location;
path_location = '..\data\case012_location.txt';
time = importdata('..\data\case012_toa.txt');
options = optimset('GradObj','on');
len = length(time(:,1));
global ti;
global mk;
ti = time;
weizhi = zeros(len,3);
for mk= 1:len
    mk
    [x,y]=fminunc('tesf',[-450 457 1 0.29],options);
    weizhi(mk,:) = x(1,1:3);

```

```

end
fid = fopen('..\results2\case012_output.txt','wt');
for i = 1: len
    fprintf(fid,'%f\t%f\t%f\n',weizhi(i,1:3));
end
fclose(fid);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function case013
clear all, close all    %%%%%%%%%%kk
global path_location;
path_location = '..\data\case013_location.txt';
time = importdata('..\data\case013_toa.txt');
options = optimset('GradObj','on');
len = length(time(:,1));
global ti;
global mk;
ti = time;
weizhi = zeros(len,3);
for mk= 1:len
    mk
    [x,y]=fminunc('tesf',[-100 200 1 0.29],options);
    weizhi(mk,:) = x(1,1:3);
end
fid = fopen('..\results2\case013_output.txt','wt');
for i = 1: len
    fprintf(fid,'%f\t%f\t%f\n',weizhi(i,1:3));
end
fclose(fid);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function case014
clear all, close all    %%%%%%%%%%kk
global path_location;
path_location = '..\data\case014_location.txt';
time = importdata('..\data\case014_toa.txt');
options = optimset('GradObj','on');
len = length(time(:,1));
global ti;
global mk;
ti = time;
weizhi = zeros(len,3);
for mk= 1:len
    mk

```

```

        [x,y]=fminunc('tesf',[-450 457 1 0.29],options);
        weizhi(mk,:) = x(1,1:3);
    end
    fid = fopen('..\results2\case014_output.txt','wt');
    for i = 1: len
        fprintf(fid,'%f\t%f\t%f\n',weizhi(i,1:3));
    end
    fclose(fid);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function case015
clear all, close all %%%%%%%%%%kk
global path_location;
path_location = '..\data\case015_location.txt';
time = importdata('..\data\case015_toa.txt');
options = optimset('GradObj','on');
len = length(time(:,1));
global ti;
global mk;
ti = time;
weizhi = zeros(len,3);
for mk= 1:len
    mk
    [x,y]=fminunc('tesf',[-450 457 1 0.29],options);
    weizhi(mk,:) = x(1,1:3);
end
fid = fopen('..\results2\case015_output.txt','wt');
for i = 1: len
    fprintf(fid,'%f\t%f\t%f\n',weizhi(i,1:3));
end
fclose(fid);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function case006
clear all, close all %%%%%%%%%%kk
global path_location;
path_location = '..\data\case016_location.txt';
time = importdata('..\data\case016_toa.txt');
options = optimset('GradObj','on');
len = length(time(:,1));
global ti;
global mk;
ti = time;
weizhi = zeros(len,3);

```

```

for mk= 1:len
    mk
    [x,y]=fminunc('tesf',[-450 457 1 0.29],options);
    weizhi(mk,:) = x(1,1:3);
end
fid = fopen('..\results2\case016_output.txt','wt');
for i = 1: len
    fprintf(fid,'%f\t%f\t%f\n',weizhi(i,1:3));
end
fclose(fid);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function case017
clear all, close all    %%%%%%%%%%kk
global path_location;
path_location = '..\data\case017_location.txt';
time = importdata('..\data\case017_toa.txt');
options = optimset('GradObj','on');
len = length(time(:,1));
global ti;
global mk;
ti = time;
weizhi = zeros(len,3);
for mk= 1:len
    mk
    [x,y]=fminunc('tesf',[-450 457 1 0.29],options);
    weizhi(mk,:) = x(1,1:3);
end
fid = fopen('..\results2\case017_output.txt','wt');
for i = 1: len
    fprintf(fid,'%f\t%f\t%f\n',weizhi(i,1:3));
end
fclose(fid);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function case018
clear all, close all    %%%%%%%%%%kk
global path_location;
path_location = '..\data\case018_location.txt';
time = importdata('..\data\case018_toa.txt');
options = optimset('GradObj','on');
len = length(time(:,1));
global ti;
global mk;

```

```

ti = time;
weizhi = zeros(len,3);
for mk= 1:len
    mk
    [x,y]=fminunc('tesf',[-450 457 1 0.29],options);
    weizhi(mk,:) = x(1,1:3);
end
fid = fopen('..\results2\case008_output.txt','wt');
for i = 1: len
    fprintf(fid,'%f\t%f\t%f\n',weizhi(i,1:3));
end
fclose(fid);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function case019
clear all, close all %%%%%%%%%%kk
global path_location;
path_location = '..\data\case019_location.txt';
time = importdata('..\data\case019_toa.txt');
options = optimset('GradObj','on');
len = length(time(:,1));
global ti;
global mk;
ti = time;
weizhi = zeros(len,3);
for mk= 1:len
    mk
    [x,y]=fminunc('tesf',[-450 457 1 0.29],options);
    weizhi(mk,:) = x(1,1:3);
end
fid = fopen('..\results2\case019_output.txt','wt');
for i = 1: len
    fprintf(fid,'%f\t%f\t%f\n',weizhi(i,1:3));
end
fclose(fid);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function case020
clear all, close all %%%%%%%%%%kk
global path_location;
path_location = '..\data\ case020_location.txt';
time = importdata('..\data\ case020_toa.txt');
options = optimset('GradObj','on');
len = length(time(:,1));

```

```

global ti;
global mk;
ti = time;
weizhi = zeros(len,3);
for mk= 1:len
    mk
    [x,y]=fminunc('tesf',[-200 200 1 0.29],options);
    weizhi(mk,:) = x(1,1:3);
end
fid = fopen('..\results2\case020_output.txt','wt');
for i = 1: len
    fprintf(fid,'%f\t%f\t%f\n',weizhi(i,1:3));
end
fclose(fid);
end

```

附录 2

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [f,g]= testf2(x)
global path_location;  %'..\data\sample_case002_location.txt'
c1 = importdata(path_location);
len1 = length(c1(:,1));
global ti;
global mk;
c = ti(mk,:);
k1=[0.029277507804620 1.543344122922481e-09];
for i=1:len1
    th(i)=(c(i)-(c(i)*x(3)+k1(2)))*3*10^8;

    th0(i)=c(i)*3*10^8;
    % %%%%%%%%%
    y0(i)=(th(i)-sqrt((x(1)-c1(i,1))^2+(x(2)-c1(i,2))^2))^2;
    g0(i)=2*(th(i)-sqrt((x(1)-c1(i,1))^2+(x(2)-c1(i,2))^2))*(-2*
(x(1)-c1(i,1))/sqrt((x(1)-c1(i,1))^2+(x(2)-c1(i,2))^2));
    g1(i)=2*(th(i)-sqrt((x(1)-c1(i,1))^2+(x(2)-c1(i,2))^2))*(-2*
(x(2)-c1(i,2))/sqrt((x(1)-c1(i,1))^2+(x(2)-c1(i,2))^2));

    g4(i)=2*(th(i)-sqrt((x(1)-c1(i,1))^2+(x(2)-c1(i,2))^2))*3*10^8*(-c(i)
);
end
f=sum(y0);
g=[sum(g0);sum(g1);sum(g4)];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function case021
clear all, close all

```

```

global path_location;
path_location = '..\data\case021_location.txt';
time = importdata('..\data\case021_toa.txt');
options = optimset('GradObj','on');
len = length(time(:,1));
global ti;
global mk;
ti = time;
weizhi = zeros(len,2);
for mk= 1:len
    mk
    [x,y]=fminunc('testf2',[-50 200 0.29],options);
    weizhi(mk,:) = x(1,1:2);
end
fid = fopen('..\results2\case021_output.txt','wt');
for i = 1: len
    fprintf(fid,'%f\t%f\n',weizhi(i,1:2));
end
fclose(fid);
plot(weizhi(:,1),weizhi(:,2),'.');
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function case022
clear all, close all
global path_location;
path_location = '..\data\case022_location.txt';
time = importdata('..\data\case022_toa.txt');
options = optimset('GradObj','on');
len = length(time(:,1));
global ti;
global mk;
ti = time;
weizhi = zeros(len,2);
for mk= 1:len
    mk
    [x,y]=fminunc('testf2',[-50 200 0.29],options);
    weizhi(mk,:) = x(1,1:2);
end
fid = fopen('..\results2\case022_output.txt','wt');
for i = 1: len
    fprintf(fid,'%f\t%f\n',weizhi(i,1:2));
end
fclose(fid);
plot(weizhi(:,1),weizhi(:,2),'.');

```

```

end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function case023
clear all, close all
global path_location;
path_location = '..\data\case023_location.txt';
time = importdata('..\data\case023_toa.txt');
options = optimset('GradObj','on');
len = length(time(:,1));
global ti;
global mk;
ti = time;
weizhi = zeros(len,2);
for mk= 1:len
    mk
    [x,y]=fminunc('testf2',[-50 200 0.29],options);
    weizhi(mk,:) = x(1,1:2);
end
fid = fopen('..\results2\case023_output.txt','wt');
for i = 1: len
    fprintf(fid,'%f\t%f\n',weizhi(i,1:2));
end
fclose(fid);
plot(weizhi(:,1),weizhi(:,2),'.');
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function case024
clear all, close all
global path_location;
path_location = '..\data\case024_location.txt';
time = importdata('..\data\case024_toa.txt');
options = optimset('GradObj','on');
len = length(time(:,1));
global ti;
global mk;
ti = time;
weizhi = zeros(len,2);
for mk= 1:len
    mk
    [x,y]=fminunc('testf2',[-50 200 0.29],options);
    weizhi(mk,:) = x(1,1:2);
end
fid = fopen('..\results2\case024_output.txt','wt');
for i = 1: len

```



```

        fprintf(fid, '%f\t%f\n', weizhi(i,1:2));
    end
    fclose(fid);
    plot(weizhi(:,1), weizhi(:,2), '.');
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function case025
clear all, close all
global path_location;
path_location = '..\data\case025_location.txt';
time = importdata('..\data\case025_toa.txt');
options = optimset('GradObj', 'on');
len = length(time(:,1));
global ti;
global mk;
ti = time;
weizhi = zeros(len,2);
for mk= 1:len
    mk
    [x,y]=fminunc('testf2', [-50 200 0.29], options);
    weizhi(mk,:) = x(1,1:2);
end
fid = fopen('..\results2\case025_output.txt', 'wt');
for i = 1: len
    fprintf(fid, '%f\t%f\n', weizhi(i,1:2));
end
fclose(fid);
plot(weizhi(:,1), weizhi(:,2), '.');
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

附加 4

附录 4 中的函数都会用到附录 1 中的 testf 函数

```

function case026
clear all, close all %%%%%%%%%kk
global path_location;
path_location = '..\data\case026_location.txt';
time = importdata('..\data\case026_toa.txt');
options = optimset('GradObj', 'on');
len = length(time(:,1));
global ti;
global mk;
ti = time;
weizhi = zeros(len,3);
for mk= 1:len

```

```

        mk
        [x,y]=fminunc('tesf',[-450 457 1 0.29],options);
        weizhi(mk,:) = x(1,1:3);
    end
    fid = fopen('..\results2\case026_output.txt','wt');
    for i = 1: len
        fprintf(fid,'%f\t%f\t%f\n',weizhi(i,1:3));
    end
    fclose(fid);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function case027
clear all, close all %%%%%%%%%%kk
global path_location;
path_location = '..\data\case027_location.txt';
time = importdata('..\data\case027_toa.txt');
options = optimset('GradObj','on');
len = length(time(:,1));
global ti;
global mk;
ti = time;
weizhi = zeros(len,3);
for mk= 1:len
    mk
    [x,y]=fminunc('tesf',[-450 457 1 0.29],options);
    weizhi(mk,:) = x(1,1:3);
end
fid = fopen('..\results2\case027_output.txt','wt');
for i = 1: len
    fprintf(fid,'%f\t%f\t%f\n',weizhi(i,1:3));
end
fclose(fid);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function case028
clear all, close all %%%%%%%%%%kk
global path_location;
path_location = '..\data\case028_location.txt';
time = importdata('..\data\case028_toa.txt');
options = optimset('GradObj','on');
len = length(time(:,1));
global ti;
global mk;
ti = time;

```

```

weizhi = zeros(len,3);
for mk= 1:len
    mk
    [x,y]=fminunc('tesf',[-450 457 1 0.29],options);
    weizhi(mk,:) = x(1,1:3);
end
fid = fopen('..\results2\case028_output.txt','wt');
for i = 1: len
    fprintf(fid,'%f\t%f\t%f\n',weizhi(i,1:3));
end
fclose(fid);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function case029
clear all, close all %%%%%%%%%%kk
global path_location;
path_location = '..\data\case029_location.txt';
time = importdata('..\data\case029_toa.txt');
options = optimset('GradObj','on');
len = length(time(:,1));
global ti;
global mk;
ti = time;
weizhi = zeros(len,3);
for mk= 1:len
    mk
    [x,y]=fminunc('tesf',[-450 457 1 0.29],options);
    weizhi(mk,:) = x(1,1:3);
end
fid = fopen('..\results2\case029_output.txt','wt');
for i = 1: len
    fprintf(fid,'%f\t%f\t%f\n',weizhi(i,1:3));
end
fclose(fid);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function case030
clear all, close all %%%%%%%%%%kk
global path_location;
path_location = '..\data\case030_location.txt';
time = importdata('..\data\case030_toa.txt');
options = optimset('GradObj','on');
len = length(time(:,1));
global ti;

```

```

global mk;
ti = time;
weizhi = zeros(len,3);
for mk= 1:len
    mk
    [x,y]=fminunc('tesf',[-450 457 1 0.29],options);
    weizhi(mk,:) = x(1,1:3);
end
fid = fopen('..\results2\case030_output.txt','wt');
for i = 1: len
    fprintf(fid,'%f\t%f\t%f\n',weizhi(i,1:3));
end
fclose(fid);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function q4 %%%%%更改路径即可计算其他case
loc = importdata('..\data\case030_location.txt');
move = importdata('..\results2\case030_output.txt');
loc_num = length(loc(:,1));
move_num = length(move(:,1));
num = 0;
fid = fopen('..\results2\output_case_030.txt','wt');
for i = 1:move_num
    cnum = 0;
    for j = 1 : loc_num
        if( sqrt(sum((loc(j,:)-move(i,:)).^2))<=200)
            cnum = cnum +1;
        end
    end
    if(cnum >5)
        num = num+1;
        fprintf(fid,'%f    %f    %f\n',move(i,:));
    else
        fprintf(fid,'%c%c%c%c    %c%c%c%c    %c%c%c%c\n', ['xnan' 'ynan'
'znan']);
    end
end
end
fclose(fid);
lambda = num/move_num

```