

参赛密码 _____
(由组委会填写)

第十二届“中关村青联杯”全国研究生 数学建模竞赛

学 校 重庆邮电大学

参赛队号 10617003

1.张瑜

队员姓名 2.谭瑞

3.李林乐

参赛密码 _____
(由组委会填写)



TCT 交控科技
Traffic Control Technology



第十二届“中关村青联杯”全国研究生 数学建模竞赛

题目 移动通信中的无线信道“指纹”特征建模

摘要：

本文为研究无线信道“指纹”特征，首先，从图像观测模型和统计分析模型获得的基本“指纹”特征，能够对三种场景进行区分。然后利用统计分析模型对问题二的场景进行识别，且能够正确识别。再运用滑动窗口模型对问题三的路径进行区域划分，最后，利用等高线“指纹”特征对样本路段进行了归类并论证了归类的正确性。

针对问题一，通过原始测量数据作幅值的散点图，并观察其分布情况，分别建立无线信道“指纹”模型：图像观测模型和统计学分析模型。由图像观测模型，建立幅值 $z = f(n, k)$ 关于测试时刻 n 和采样点数 k 的空间三维空间散点图和曲面图，得出关于图形的直观特征。该模型能够直观的对三种场景进行合理区分。由统计学分析模型得到每种场景信号能量无量纲化的方差特征，再对数据进行聚类分析并进行无量纲化处理，得出无量纲化半径方差特征，该模型也能够对三种场景进行合理区分。

针对问题二，分别采用问题一中建立的两个“指纹”模型，提取“数据包 2”中两个样本的特征，再通过最小二乘法和判别分析法（Bayes 准则和 Fisher 准则）来识别“数据包 2”两个样本的特征，能够识别出“数据包 2”提供的 Test1ForSence 数据采集于问题一的场景一，Test2ForSence 数据采集于问题一的场景二，并得到对应的识别率。还可将问题一的三种场景中前 4 个 Test 作为训练样本，用剩下的 Test 作为测试样本，建立神经网络模型，可得到与前面相同的识别结果。

针对问题三，采用问题一中的统计学分析模型，对“数据包 3”的数据进行分段，再改进得到滑动窗口模型并对划分结果进行分析，可得到较合理的“指纹”区

域划分为：7 段 4 个场景 (0m, 12.71m, 16.95m, 42.37m, 63.56m, 67.79m, 116.95m, 150m)。最后利用等高线“指纹”特征可验证划分的“指纹”区域是正确的。

针对问题四，首先采用问题一中统计分析模型提取“数据包 4”中两个样本的“指纹”特征，然后利用滑动窗口匹配方法，判断出样本一不属于“数据包 3”中的路段，样本二属于“数据包 3”中场景四的 8900~9100 路段，得到较小的误判距离为 $3.9069e-18$ 。最后，利用等高线“指纹”特征对模型计算结果进行比较，验证了模型的正确性。

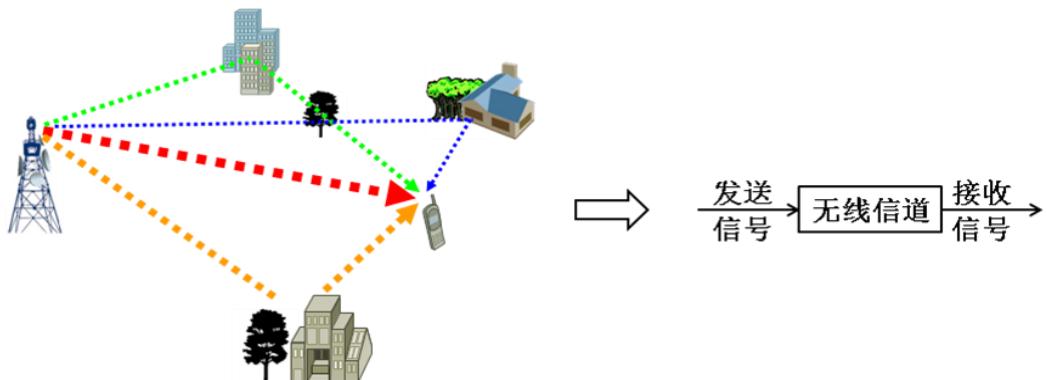
关键词：等高线“指纹”特征 统计分析 BP 场景识别 滑动窗口

一、问题重述

1、背景介绍

移动通信产业一直以惊人的速度迅猛发展，已成为带动全球经济发展的主要高科技产业之一，并对人类生活及社会发展产生了巨大的影响。在移动通信中，发送端和接收端之间通过电磁波来传输信号，我们可以想象两者之间有一些看不见的电磁通路，并把这些电磁通路称为无线信道。无线信道与周围的环境密切相关，不同环境下的无线信道具有一些差异化的特征。如何发现并提取这些特征并将其应用于优化无线网络，是当前的一个研究热点。类比人类指纹，我们将上述无线信道的差异化的特征称为无线信道“指纹”。无线信道“指纹”特征建模，就是在先验模型和测试数据的基础上，提取不同场景或不同区域内无线信道的差异化的特征，进而分析归纳出“指纹”的“数学模型”，并给出清晰准确的“数学描述”。

在典型的无线信道中，电磁波的传输不是单一路径的，而是由许多因散射（包括反射和衍射）而形成的路径所构成的。由于电磁波沿各条路径的传播距离不同，因此相同发射信号经由各条路径到达接收端的时间各不相同，即多径的时延之间有差异。此外各条路径对相同发射信号造成的影响各不相同，即多径的系数之间有差异。如左下图所示：



工程上，考虑到多径系数及多径时延的影响，在保证精度的前提下，可以用“离散线性系统”为无线信道建模。需要注意的是，该模型中的信号及多径系数均为复数。

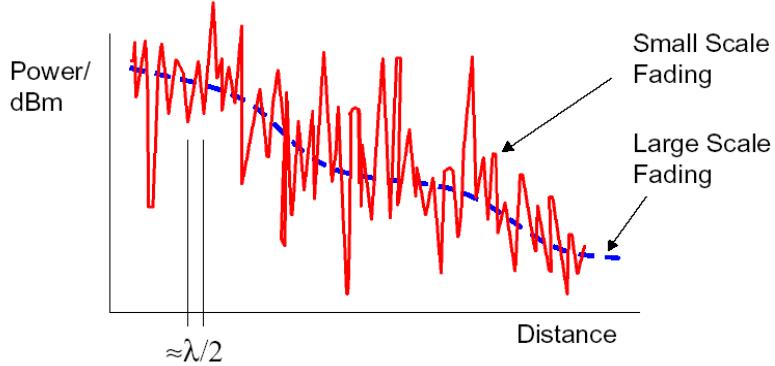
理想信道测量可以理解为获取该系统的单位序列响应，即获取单位脉冲“ $\delta[k]$ ”经无线信道传输后被接收到的信号，如右上图所示。上述理想信道测量的结果用公式表述如下：

$$h[k] = \sum_{l=0}^{L-1} h_l \cdot \delta[k - \tau_l], k = 0, 1, \dots, K-1, K \geq \max x_l \{\tau_l\}$$

$$\delta[k] = \begin{cases} 0, & k \neq 0 \\ 1, & k = 0 \end{cases}$$

其中，“ k ”为离散信号的样点标识，这里假设共有“ K ”个样点；“ L ”是当前时刻的路径总数；“ h_l ”为当前时刻第 l 条路径上的信道系数，通常是复数；“ τ_l ”为当前时刻第 l 条路径的时延，且已折算成样点数，即延迟了“ τ_l ”个样点。显然，复信号

“ $h[k]$ ”给出了当前时刻的完整信道。需要强调的是，上述各个参数，包括“ h_l ”、“ τ_l ”和“ L ”都会随着时间而变化，即各个参数具有时变性。相应地，“ $h[k]$ ”的功率在信号波长“ λ ”的量级上会出现时而加强时而减弱的快速变化，称之为多径衰落或小尺度衰落。同时，快速变化的功率，其平均值也会出现缓慢的变化，这主要是由于周围环境或气象条件的改变而引起的，称之为阴影衰落或大尺度衰落。两种衰落特征如下图所示：



上述理想信道测量的结果“ $h[k]$ ”是无法直接获取的。因为在真实无线通信系统中，为了改善信号的传输质量，通常需要在系统的发射端和接收端，各增加一个滤波器。所有滤波器在真实信道测量中的影响，可以等效地用函数“ $g[k]$ ”来表示。此时信道测量的结果为：

$$r[k] = \sum_{m=0}^{M-1} h[k-m] \cdot g[m], k = 0, 1, \dots, K-1$$

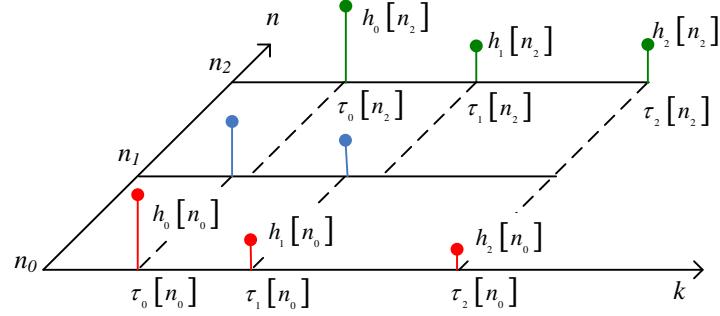
其中，“ M ”为滤波器的长度，即“ $g[k]$ ”的样点数。考虑到信道的时变性以及实测中引入了噪声，不同时刻的真实信道测量结果及其对应的无线信道分别为：

$$r[k, n] = \sum_{m=0}^{M-1} h[k-m, n] \cdot g[m] + u[k, n], k = 0, 1, \dots, K-1, n = 0, 1, \dots, N-1$$

$$h[k, n] = \sum_{l=0}^{L-1} h_l[n] \cdot \delta[k - \tau_l[n]]$$

其中，“ n ”表示测试的样本标识，对应测试时刻，这里假设共有“ N ”个样本；“ $h_l[n]$ ”表示“ n ”时刻第 l 条路径上的信道系数，通常是复数；“ $\tau_l[n]$ ”表示“ n ”时刻第 l 条路径延迟的样点数；“ $u[k, n]$ ”表示“ n ”时刻第“ k ”个测试样本上引入的复高斯白噪声；“ $r[k, n]$ ”表示“ n ”时刻单位脉冲依次经发送滤波器、信道和接收滤波器后的

实际接收信号，是 k, n 的二元函数。显然，我们可以从“ $r[k, n]$ ”中获取完整的时变信道“ $h[k, n]$ ”。为了便于理解，下图给出了不同时刻下无线信道的示意图，同时也给出了样本标识“ n ”和样点标识“ k ”的相互关系。容易发现，不同时刻下多径的条数、时延以及系数值都有可能发生变化。直观上，变化的参数都含有一些场景化的特征，即上述不同参数的变化可能存在一定的规律。



基于上述或更多的参数及特征，一个场景或一定区域内的无线信道可能存在一定的“指纹”。利用所定义的“指纹”，进行场景的分析以及无线网络的优化，具有重要意义。我们希望利用真实信道测量结果分析建立信道的“指纹”模型。该模型可以是一个参数或多个参数（例如多径数目、多径时延、时延扩展、Doppler 扩展、是否有直达径等）的组合或函数，也可以是某种图形化的描述；可能是时间域的，也可能是频率域的；可能是确定性的，也可能是统计意义上的。关键是所建立“指纹”模型能够从统计意义上有效区分出某个场景或区域。

2、待研究的问题

问题 1：根据“数据包 1”提供了三种场景的真实信道测量结果。采用数学建模特别是特征提取的方法，对所提供的样本加以分析，给出无线信道“指纹”的模型。在此基础上，给出简洁而明确的“评价指标”，用于分析验证所建模“指纹”合理有效。所提取的特征或所建立的模型最好有一定的物理意义。

问题 2：根据“数据包 2”提供的二个真实信道测量结果。基于“问题 1”中所提供的三个场景，以及所建立的“指纹”模型和“评价指标”，采用数学的方法，识别出此处提供的二个样本分别属于哪个场景。且对识别的结果进行合理的分析。

问题 3：根据“数据包 3”提供的一段实际路段的真实信道测量结果。在该路段以 3km/h 的速度步行近 150m 的测试距离。求出该路段可能包含不同的场景或环境，对应不同的“指纹”特征。基于上述测量结果，先自行分段，采用“问题 1”中的建模方法，给出分段的“指纹”分析。在分段分析的基础上，对所提取的“指纹”进行合理分类。通过对不同段的分析结果，最终决定该路段可以依“指纹”划分为多少个区域。详细说明“区域划分”的思路及过程，并对划分结果进行合理的分析。

问题 4：根据“数据包 4”提供的两个真实信道测量结果。基于“问题 3”中的“区域划分”和“问题 1”中的“评价指标”，首先判断此处提供的二个样本是否采集自“问题 3”中所提供的路段。对于已判断出的采自上述路段的样本，请识别其对应于“问题 3”中的哪一块区域。对所识别的结果进行合理的分析并详细说明样本判别的思路和方法，以及得出结论的过程。

二、问题分析

对于问题一，可用“数据包 1”中三种场景真实信道的原始数据作图，并观察其分布情况，分别建立无线信道“指纹”模型：图像观测模型和统计学分析模型。由图像观测模型，建立幅值 $z = f(n, k)$ 关于测试时刻 n 和采样点数 k 的空间三维空间散点图和曲面图^[1]，该模型能够直观的对三种场景进行合理区分。由统计学分析模型得到每种场景信号能量无量纲化的方差特征，再对数据进行聚类分析并进行无量纲化处理得到无量纲化半径方差，进而得到方差变化率的特征，且该模型也能够对三种场景进行合理区分。

对于问题二，分别采用问题一中建立的两个“指纹”模型，提取“数据包 2”中两个样本的特征，再通过最小二乘法和判别分析法（Bayes 准则和 Fisher 准则）来识别“数据包 2”的两个样本，能够识别出“数据包 2”提供的两个样本分别属于问题一中三种场景的哪个场景，并得到对应的识别率。还可将问题一中三种场景的前 4 个 Test 作为训练样本，用剩下的 Test 作为测试样本，建立神经网络模型，同样可以对不同场景进行识别。

对于问题三，由于“数据包 3”中数据较多，为降低问题复杂度，采用问题一中建立的统计学分析模型，可先对其数据进行分段，再对统计学分析模型进行改进得到滑动窗口模型并对划分结果进行分析，可得到较合理的“指纹”区域划分。最后利用等高线“指纹”特征可验证划分的“指纹”区域是合理的。

对于问题四，为了识别“数据包 4”提供的两个样本分别属于“问题 3”中的哪一块区域，采用问题一中的统计分析模型提取这里两个样本的“指纹”特征，通过滑动窗口模型匹配“数据包 4”中两个样本对应于“数据包 3”中的最佳位置，并得到较小的误判距离。同时，利用等高线“指纹”特征对模型计算结果进行比对，判断模型的合理性。

三、模型假设与基本符号说明

2.1 模型假设

-
- 1、不考虑其他信号对测量信号的干扰；
 - 2、不同场景所用到的滤波器相同；
 - 3、短时间内气候不变；
 - 4、不考虑基站间信号的重叠与空缺；
 - 5、检测点不迂回运动。
-

2.2 基本符号说明

$B(i, k)$	i 行 k 列上的元素	$A(i)$	第 i 行所有元素的平均值
a_{ix}	$A(i)$ 的实部	b_{iy}	$A(i)$ 的虚部
a_0	实部的平均值	b_0	虚部的平均值
p_i	$A(i)$ 对应的能量	p_{\max}	最大能量值
\hat{p}_i	无量纲化后的能量	σ_p	能量无量纲化后的方差
l_{i_k}	$B(i, k)$ 到聚类中心的距离	\hat{l}_{i_k}	对 l_{i_k} 无量纲化
σ_r	l_{i_k} 的方差	θ_{i_k}	$B(i, k)$ 样点的相位
θ_{all}	θ_{i_k} 总和		

注：其余符号为方便辨认，在列示中显示。

四、模型建立与求解

4.1 问题一：无线信道“指纹”特征提取

4.1.1 问题一分析

首先，利用“数据包1”中三种场景真实信道的原始数据作图，观察其分布情况，根据图形特征对场景进行感性区分^[2]。其次，建立无线信道“指纹”模型：图像观测模型和统计学分析模型。由图像观测模型，建立幅值 $z = f(n, k)$ 关于采样时刻 n 和采样点数 k 的空间三维空间散点图和曲面图，该模型能够直观的对三种场景进行合理区分。从图中可看出，同种场景的五个 Test 分布情况具有显著性差异。因为信号发送端和接收端的距离未知，无法根据某个采样点信号的能量大小对三种场景进行区分，所以我们对某个采样点信号的能量进行无量纲化处理，得到每种场景信号能量无量纲化的方差特征和半径无量纲化方差变化率特征，这样就可以消除距离的影响，据此可建立统计学分析模型。最后，可据这两种无线信道“指纹”模型对三种场景进行合理的区分。

4.1.2 无线信道“指纹”模型的建立与求解

(1) 数据的分析与准备

根据“数据包1”中三种场景原始数据，并以其实部作为横坐标，虚部作为纵坐标作幅值散点图，如图4.1、图4.2和图4.3所示。

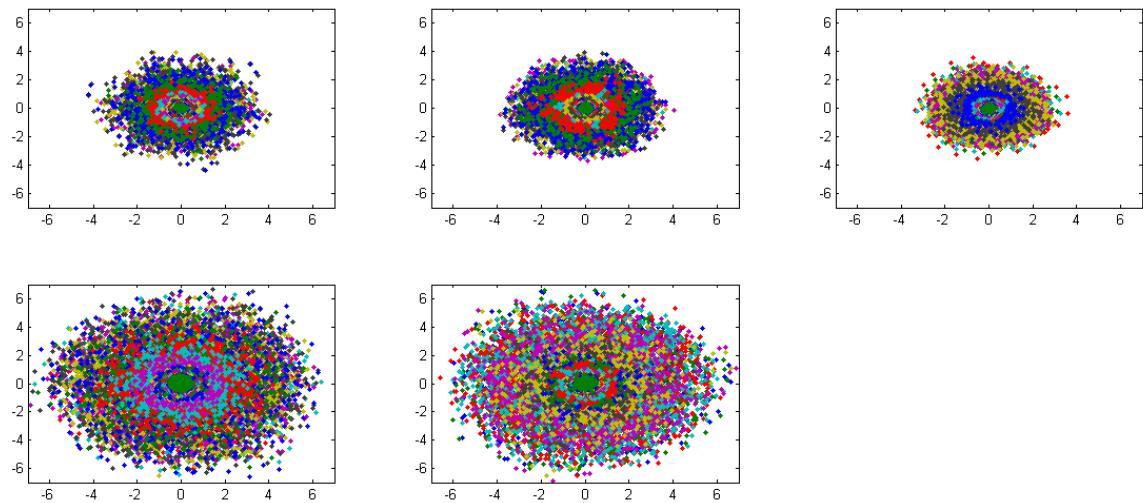


图4.1 场景一

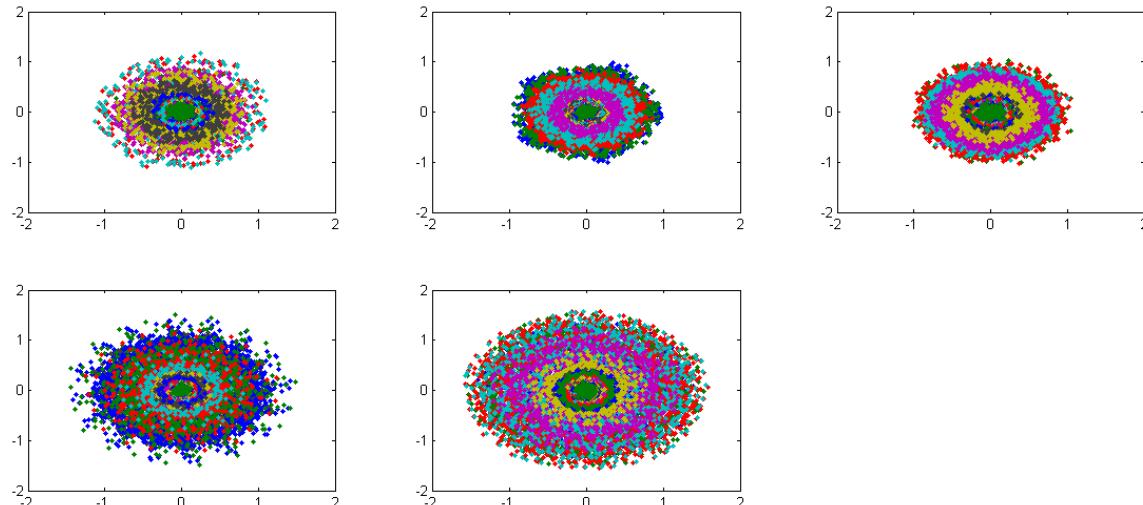


图4.2 场景二

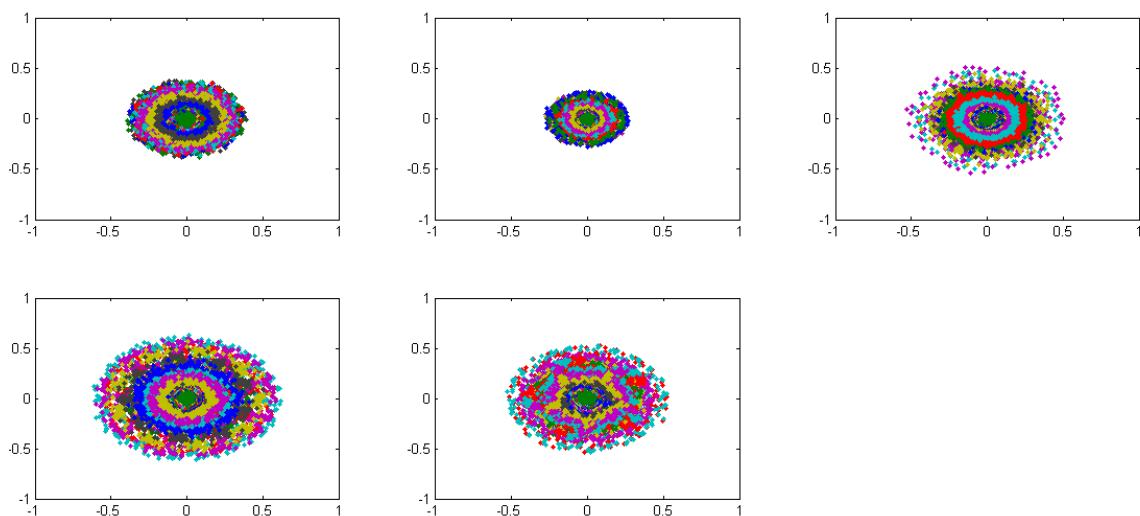


图4.3 场景三

据上述原始数据作出的三种场景幅值散点图，可看出三种场景的数据在复平面分布近似呈圆形分布，且中心点接近(0, 0)。因为数据接收端与发送端的距离是未知

的，所以通过直观半径大小的判断是不能区分三种场景。然而，同种场景5个真实信道测量结果经过处理得到的信号能量变化率和半径变化率是不相同的，据此先对其进行无量纲化，再观测其变化趋势有望区分三种场景。

(2) 图像观测模型

根据“数据包1”中三种场景原始数据，并以其信道样本数N作为X坐标，采样点数K作为Y坐标，无量纲化幅值(真实信道测量的信号实部平方与虚部平方的和开平方)作为Z坐标得到三维空间图^[3]，即5个Test接受信号分布情况，如图4.4、图4.5和图4.6所示。

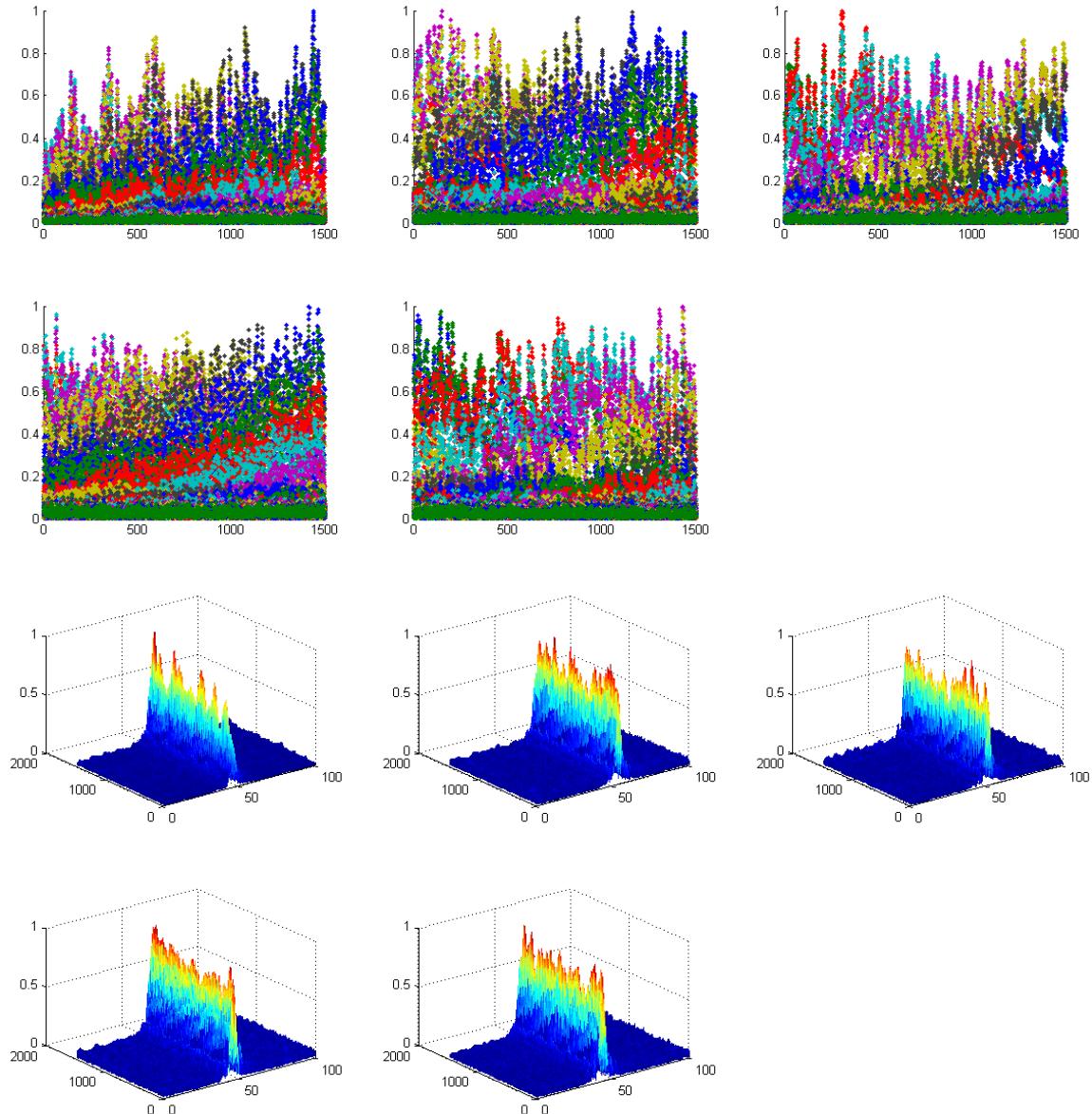


图4.4 场景一

从图4.4可以看出，各颜色平行分布且布满大半个坐标系，且空间曲面图顶端区域平行分布，没有较明显的毛刺突出。这在直观上可作为场景一“指纹”特征。

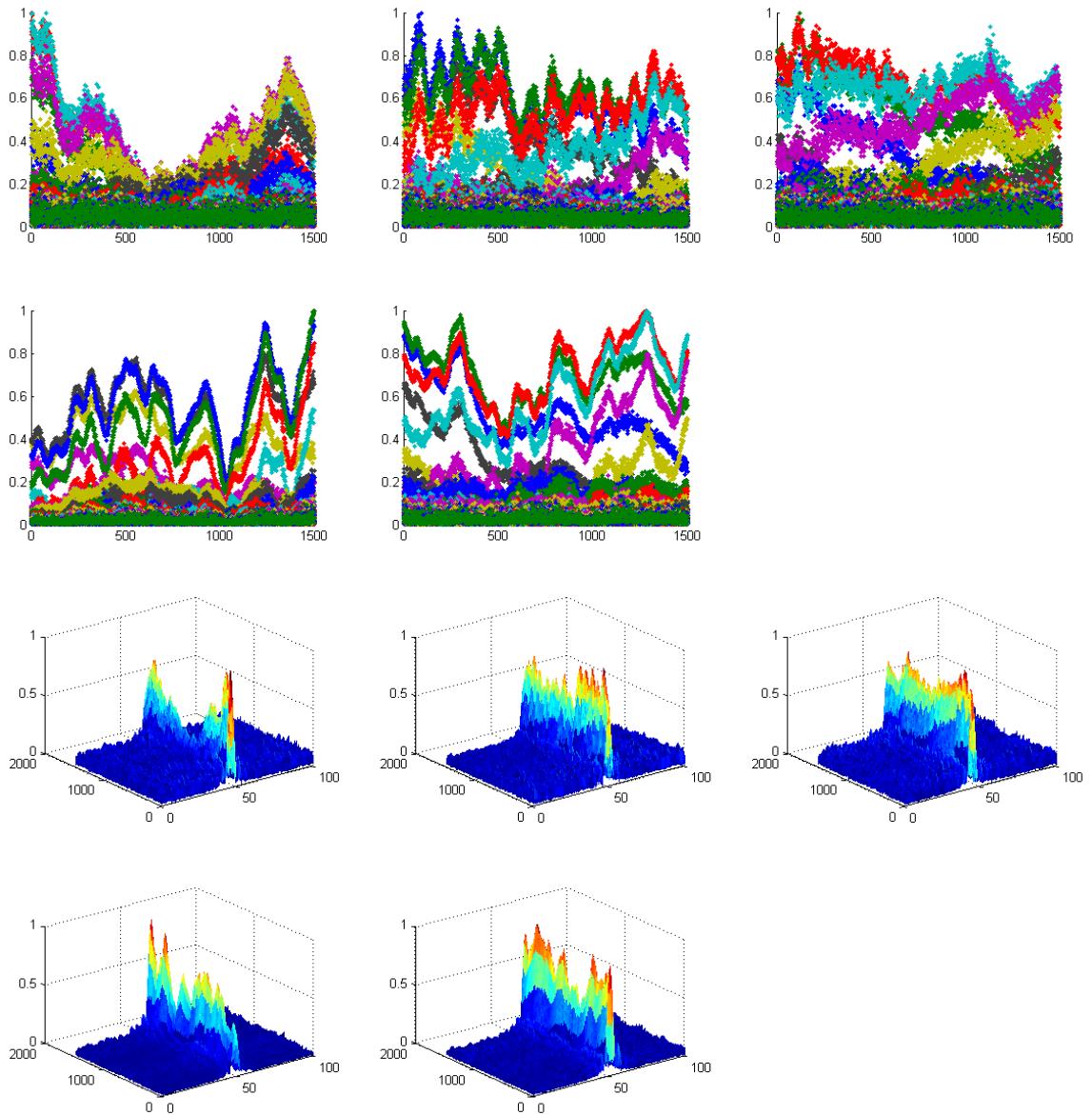


图4.5 场景二

从图4.5可以看出，各颜色线条波峰波谷频率跳动依此递减且同颜色的离散点在折线上分布较为集中。这在直观上可作为场景二“指纹”特征。

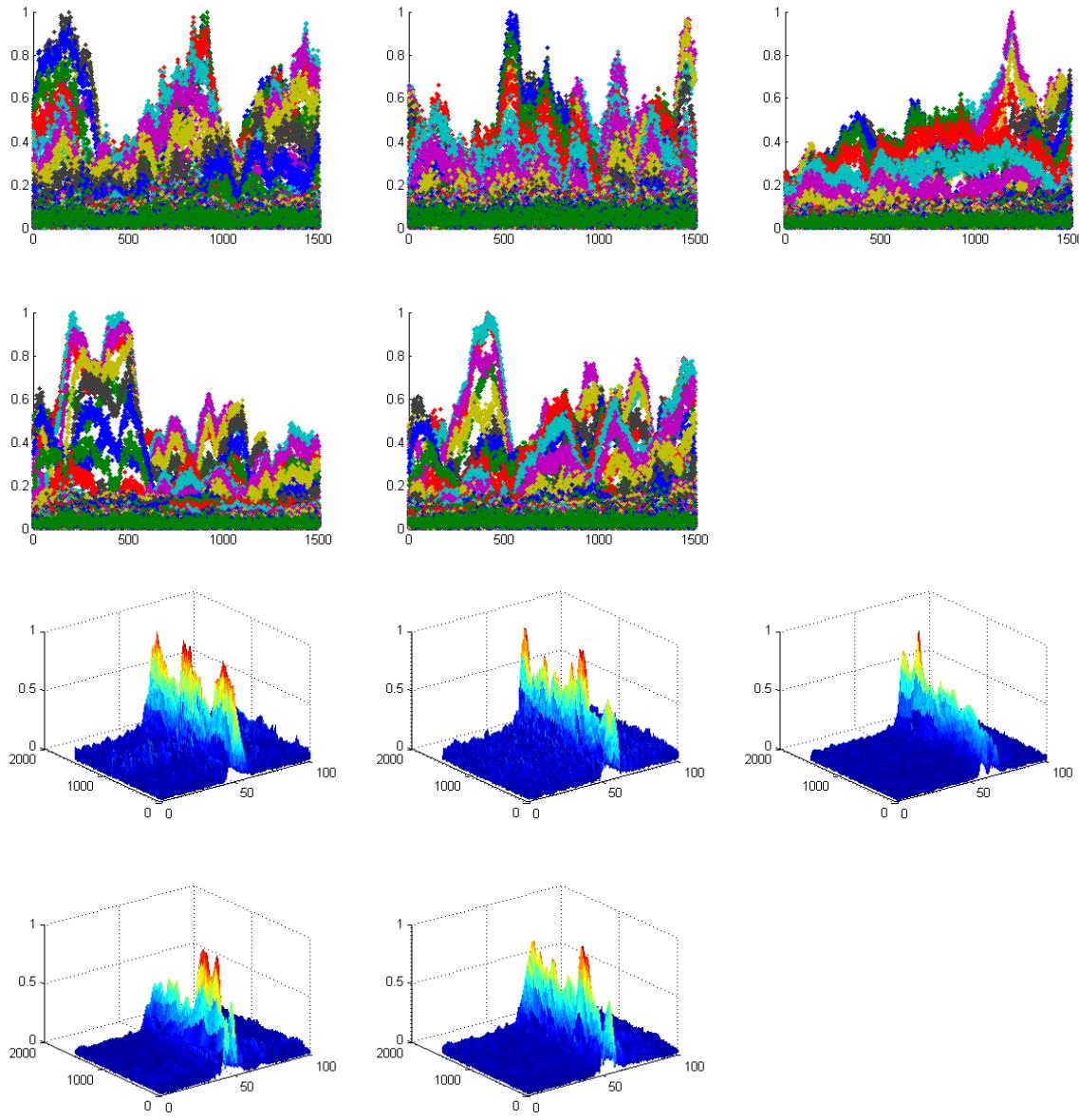


图4.6 场景三

从图4.6可以看出，图中会出现一个最大波峰且点的分布是很密集的，这在直观上可作为场景三“指纹”特征。

据上述分析，我们选择前两个场景中Test1的数据来进行说明，对场景1Test1的行向量转化为无量纲化离散化的能量点^[4]，其分布情况如图4.7所示；对场景2Test1的行向量转化为无量纲化离散化的能量点，其分布情况如图4.8所示。

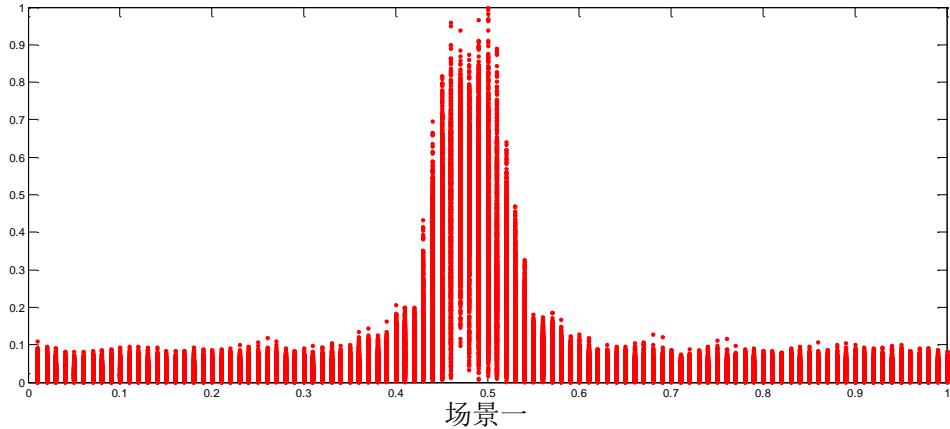


图4.7 场景一接受完整脉冲

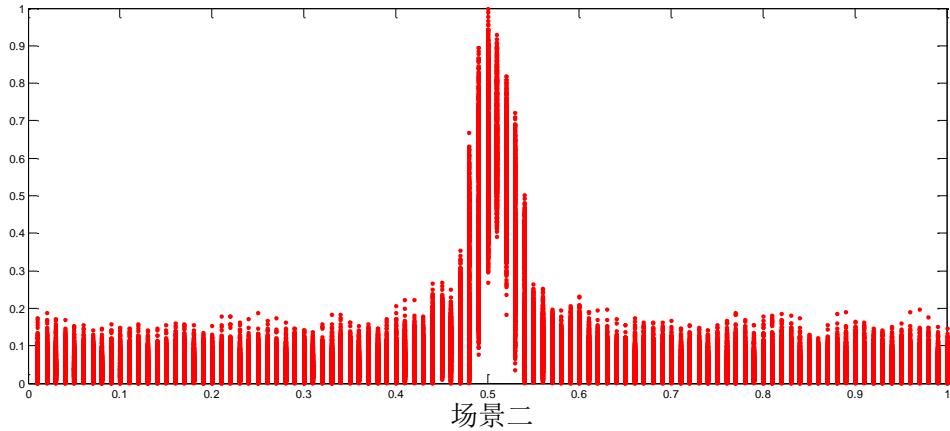


图4.8 场景二接受完整脉冲

理想情况下单条信道无损传播情况下，发送端发送一个完整的脉冲信号，于是接收端就能接收到一个完整脉冲信号。但在实际情况下考虑路径受到多径衰弱以及信道参数的影响接受端所接受到的脉冲信号是由多个不同时刻接收到离散点的加和，同种场景受到自身环境情况的影响会使得多径衰弱和信道参数在一定范围内波动^{[5][6][7]}。如图4.7和图4.8场景一和场景二所接受到无量纲化后的脉冲“胖瘦”情况是不一样的，提取该特征可作为场景的“指纹”识别依据。

(3) 统计学分析模型

1) 能量无量纲化方差作为“指纹”特征

根据 Parseval 定理，信号傅立叶变换的模再取其平方得到能量谱，即单位频率范围内包含的信号能量^[8]。因此，复平面坐标点到平面中心的距离为能量的开方，其半径大小可表征能量强弱，为去除距离对特征提取的影响，于是我们对能量进行无量纲化处理，求其方差，则该方差可作为三种场景的判别特征。模型如下：

$$A_i = \sum_{k=1}^{100} B(i, k) / 100 \quad (4.1)$$

$$P_i = a_{ix}^2 + a_{iy}^2 \quad (4.2)$$

$$\hat{P}_i = P_i / P_{\max} \quad (4.3)$$

$$\sigma_p = \sum_{i=1}^{1500} (\hat{P}_i - \hat{P})^2 / 1500 \quad (4.4)$$

其中： $A(i)$ 表示第 i 行所有元素的平均值； $B(i, k)$ 表示 i 行 k 列上的元素； i 的取值范围为 $1, 2, \dots, 1500$ ； p_i 表示 $A(i)$ 对应的能量； a_{ix} 表示 $A(i)$ 的实部； b_{iy} 表示 $A(i)$ 的虚部； p_{\max} 表示最大能量值； \hat{P}_i 表示无量纲化后的能量； σ_p 表示能量无量纲化后的方差。

通过上述模型的建立可求得无量纲化后的能量图如图 4.9、图 4.10 和图 4.11 所示，以及相应无量纲化方差特征，见表 4_1。

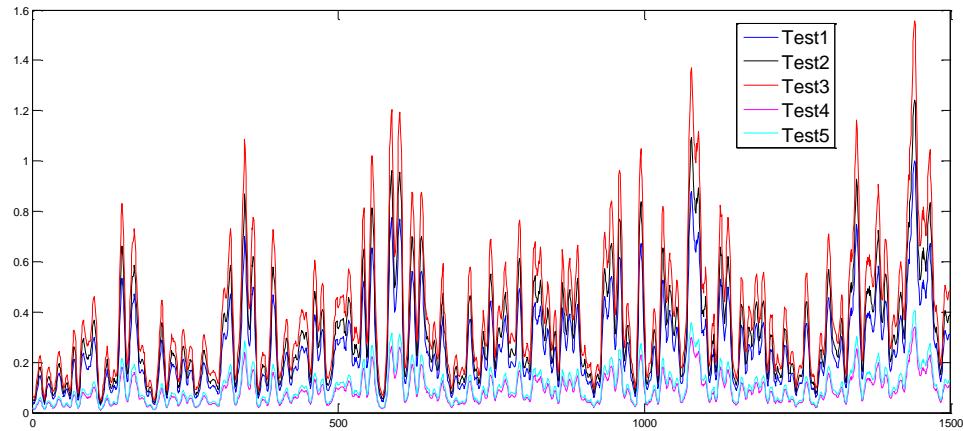


图 4.9 场景一

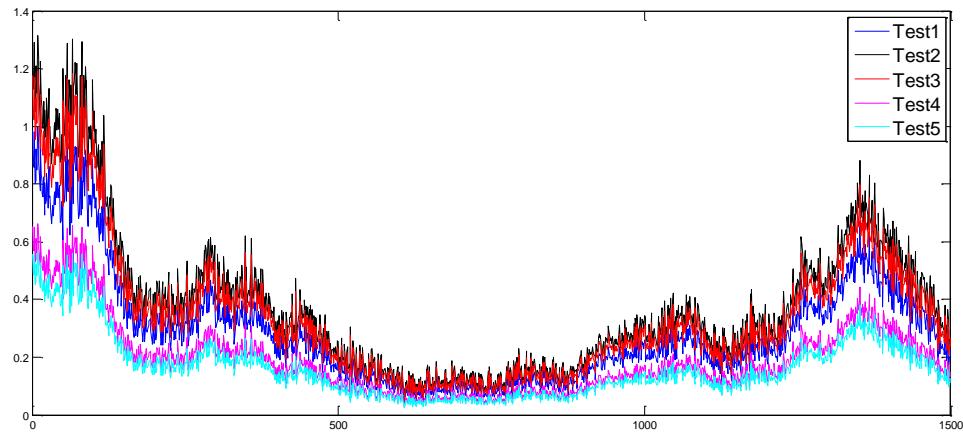


图 4.10 场景二

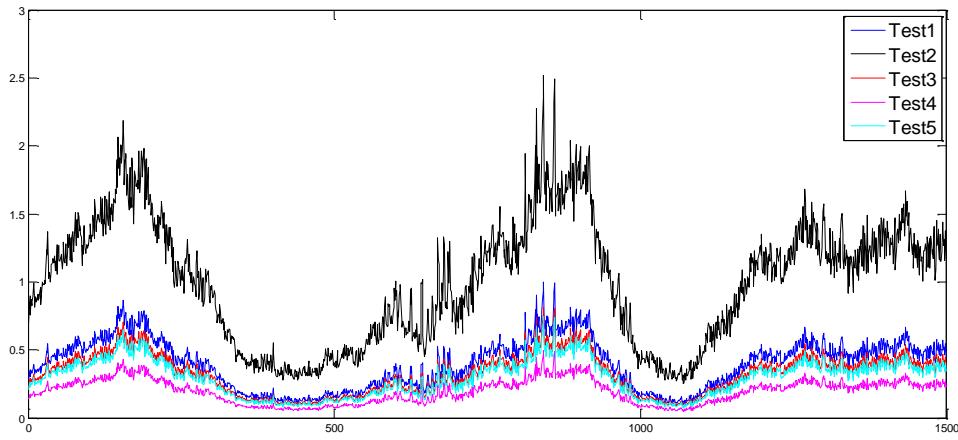


图 4.11 场景三

从图 4.9、图 4.10 和图 4.11 可直观看出三种场景下真实信道测量信号的能量谱是不一样的，场景一能量总体平均较小，变化较为平滑；场景二两端能量相对较高，中间相对较低，总体能量和场景一相当；场景三能量在信道轴上呈周期性变化，整体能量值较高。三种场景下测试数据能量的方差及变化范围如表 4_1 所示：

表 4_1 三种场景下测试数据能量的方差及变化范围

	Test1	Test2	Test3	Test4	Test5	Min	Max
场景一	0.9479	1.0839	0.7918	3.9641	3.5633	0.7918	3.9641
场景二	0.0722	0.0853	0.1020	0.1485	0.2494	0.0722	0.2494
场景三	0.0142	0.0054	0.0165	0.0313	0.0219	0.0054	0.0313

2) 半径变化率无量纲化方差作为“指纹”特征

聚类分析^{[9][10]} (cluster analyses) 作为一种定量方法，从数据分析的角度，给出一个更准确、细致的分类工具。要用数量化的方法对事物进行分类，就必须用数量化的方法描述事物之间的相似程度。一个事物常常需要用多个变量来刻画。如果对于一群有待分类的样本点需用 p 个变量描述，则每个样本点可以看成是 R^p 空间中的一个点。因此，很自然地想到可以用距离来度量样本点间的相似程度。

针对问题一的定量变量，采用 Minkowski 距离中最常用的欧式距离：

$$d_2(x, y) = \left[\sum_{k=1}^p |x_k - y_k|^2 \right]^{\frac{1}{2}} \quad (4.5)$$

因此，由原始数据分布图可求得图形中心，通过欧式公式求出各点相对中心距离，求出各列的均值并对列无量纲化处理，最后求其方差。模型如下：

$$\begin{cases} a_0 = \sum_{i=1}^{1500} \sum_{k=1}^{100} a_{ik} / 150000 \\ b_0 = \sum_{i=1}^{1500} \sum_{k=1}^{100} b_{ik} / 150000 \end{cases} \quad \text{中心} \quad (4.6)$$

$$l_{i_k} = [(a_{i_k} - a_0)^2 + (b_{i_k} - b_0)^2]^{\frac{1}{2}} \quad (4.7)$$

$$\hat{l}_{i_k} = l_{i_k} / [\sum_{i=1}^{1500} l_{i_k} / 1500] \quad \text{均值}$$
均值
(4.8)

$$\sigma_r = \sqrt{\sum_{i=1}^{1500} (\hat{l}_{i_k} - \bar{l})^2 / 150000} \quad \text{Std}$$
Std
(4.9)

其中: l_{i_k} : 表示 $B(i, k)$ 到聚类中心的距离; \hat{l}_{i_k} : 表示对 l_{i_k} 无量纲化; σ_r : 表示 l_{i_k} 的方差; i 的取值范围为 $1, 2, \dots, 1500$; k 的取值范围为 $1, 2, \dots, 100$ 。

通过上述模型提取各场景特征，并对无量纲化的半径及半径变化率作图，如图 4.12、图 4.13 和图 4.14 所示。

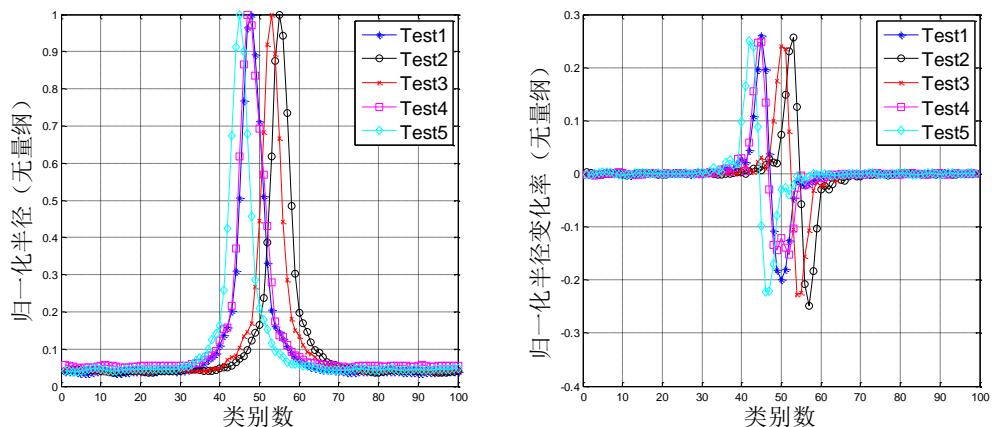


图4.12 场景一

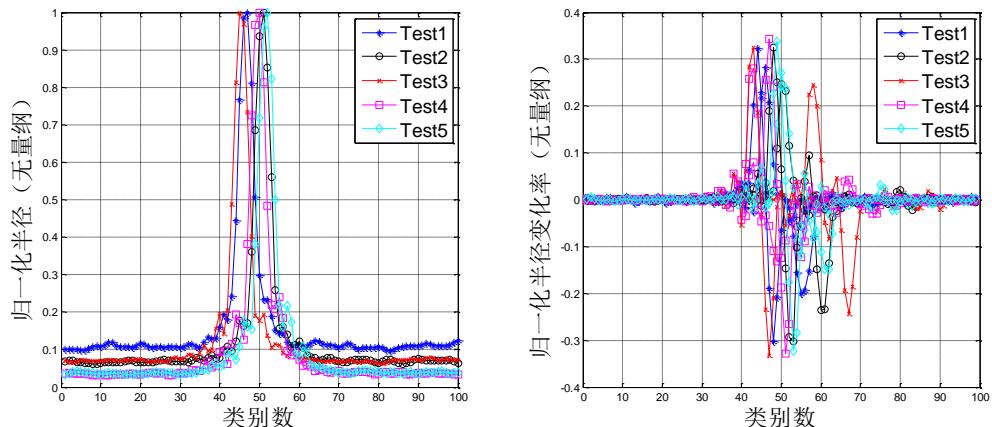


图4.13 场景二

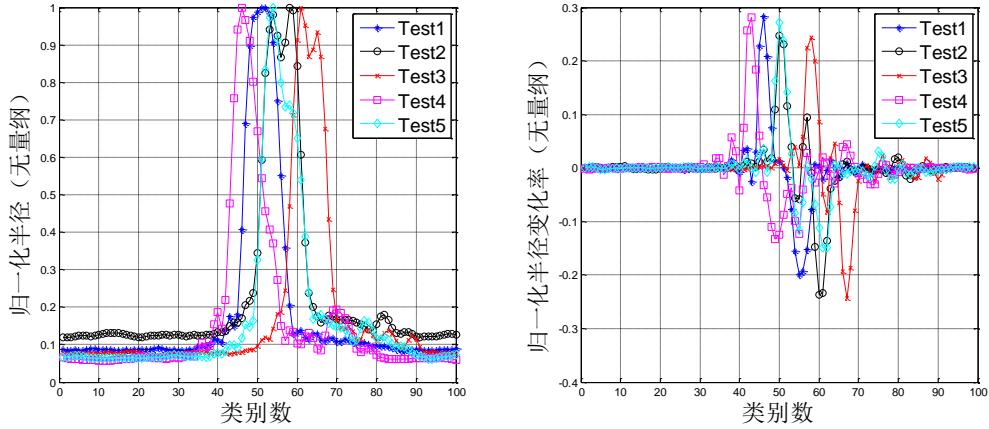


图4.14 场景三

图4.12、图4.13和图4.14可观察出无量纲化后的半径主要分布在40~70采样点范围，其中场景二的变化范围最小，大致保持在42~53；场景一变化范围其次，大致保持在40~60；场景三变化范围最大，保持在40~70。三种场景下测试数据半径变化率的方差如表4_2所示：

表4_2 三种场景下测试数据半径变化率的方差及方差变化范围

	Test1	Test2	Test3	Test4	Test5	All	Min	Max
场景一	0.3462	0.3807	0.3221	0.7696	0.6898	0.2111	0.2111	0.7696
场景二	0.0827	0.1128	0.1528	0.1497	0.2021	0.0448	0.0448	0.2021
场景三	0.0477	0.0272	0.0499	0.0624	0.0595	0.0138	0.0138	0.0624

3) 相位和作为“指纹”特征

由于多径效应，每个路径传播的信号到达接收端的时间是不一样的，这一特征在复数域的信号上表现出来为相位差^[11]。每2/3ms发送一个脉冲信号，脉冲信号通过多个路径到达接收端，导致接收端所接受到的信号存在相移。然而对单个信号而言，假定第一个通过最短路径到达接收器的相位差为零，那么剩下所接受到99个信号相位的和就能反映出该场景下总体多径时延。不同场景对应的总多径时延是不一样的，因此，总接受到信号的相位总和能作为该接收端所处场景的特征。模型如下：

$$\begin{cases} \theta_{i_k} = \arctan(b_{i_k} / a_{i_k}), a_{i_k} \geq 0 \\ \theta_{i_k} = \arctan(b_{i_k} / a_{i_k}) + \pi / 2, a_{i_k} < 0 \end{cases} \quad (4.10)$$

$$\theta_{all} = \sum_{i=1}^{1500} \sum_{k=1}^{100} \theta_{i_k} \quad (4.11)$$

其中： θ_{i_k} ：表示第 i 行 k 个元素的相位； θ_{all} ：表示总的相位和。

通过该模型计算得出结果如表4_3所示，从结果中可看出三种场景对应的相位总和，场景一和场景二之间的相位总和有较为明显交集，但对于较高数量级的数据而言该交集几乎可以忽略不计。

表4_3 三种场景下测试数据相位总和以及总和方差

	Test1	Test2	Test3	Test4	Test5	方差	Min	Max
--	-------	-------	-------	-------	-------	----	-----	-----

场景一	2.3585*e +05	2.3562*e +05	2.3607*e +05	2.3585*e +05	2.3549*e +05	221.9 958	2.3549*e +05	2.3607*e +05
场景二	2.3509*e +05	2.3536*e +05	2.3590*e +05	2.3494*e +05	2.3550*e +05	373.1 970	2.3494*e +05	2.3550*e +05
场景三	2.3668*e +05	2.3663*e +05	2.3846*e +05	2.3720*e +05	2.3635*e +05	838.0 779	2.3635*e +05	2.3720*e +05

4.1.3 结果分析

图像观测模型可从不同角度、不同方向观察图形的形状、边界、“胖瘦”和峰值等特征，据此可对三个场景进行区分。

统计分析模型中能量无量纲化方差特征可描述同一个场景中位置改变接收端接收到信号能量的变化情况，以及半径变化率无量纲化方差特征描述场景的能量分布情况。上述表和图的数据表明：场景三中位置的改变对接收到信号的能量改变影响最小，场景一最大，场景二介于两者之间。

4.2 问题二：对“数据包 2”所给样本进行场景识别

4.2.1 问题二分析

识别问题，是将目标特征与样本特征进行比对，若与某个样本特征符合程度较高则说明该目标与样本属于同一类。

为了识别出“数据包 2”中的两个样本分别属于问题一中的哪个场景，所以我们采用问题一中的统计分析模型，提取了“数据包 2”中两个样本的能量和分布特征，并通过最小二乘法和判别分析法（Bayes 准则和 Fisher 准则）来识别“数据包 2”的两个样本。

针对问题一的模型分别求取问题二中两个场景所测数据样本无量纲化后的功率谱以及聚类分析得到无量纲化后的半径如图 4.15 和图 4.16 所示。

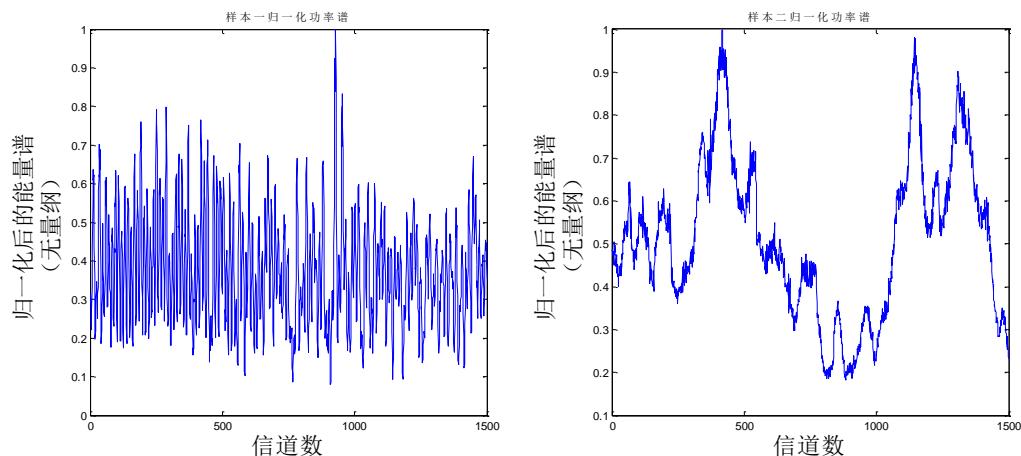


图 4.15 “数据包 2”

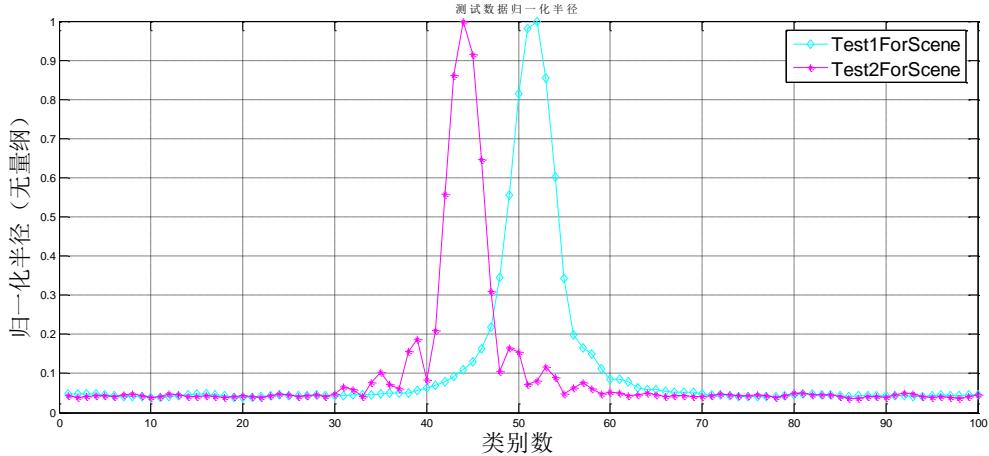


图 4.16 “数据包 2”

将“数据包 2”所给场景的数据，计算所给各场景各个时刻相位总和，与问题一中三种场景的相位范围比较可对问题二中的场景进行识别。

4.2.2 模型的建立与求解

(1) 最小二乘法求解

最小二乘法主要是通过最小化误差的平方和寻找数据的最佳函数匹配^[12]。利用最小二乘法可以简便地求得未知的数据，并使得这些求得的数据与实际数据之间误差的平方和为最小。因此，可通过最小二乘法对“数据包 2”中两个样本的特征提取，并对其识别，从而得出样本分别属于问题一三种场景中的哪一个。并计算出其识别率

最小二乘法模型：

$$\min = \sum_{i=1}^3 (z_0 - z_i)^2 \quad (4.12)$$

$$\eta = 1 - \frac{\min}{\sum_{i=1}^3 \xi_i} \quad (4.13)$$

其中： z_0 ：表示“数据包 2”数据所求得的无量纲化后的能量谱或无量纲化后的半径分布； z_i ：表示三种场景所求得无量纲化后的能量谱和无量纲化后的半径分布； ξ_i ：表示问题一中三种场景对应特征值； η ：表示识别率。

模型求解结果如表 4_4 和表 4_5 所示：

表 4_4 能量谱识别结果及识别率

无量纲化后的能量谱	“数据包 1”	识别率 (η)
Test1ForScene	场景一	80.38%
Test2ForScene	场景二	55.44%

表 4_5 半径分布识别结果及识别率

无量纲化后的半径分布	“数据包 1”	识别率 (η)
Test1ForSence	场景一	79.65%
Test2ForSence	场景二	93.15%

(2) 判别分析法求解

判别分析 (distinguish analysis) 是根据所研究的个体的观测指标来推断该个体所属类型的一种统计方法，本文使用Bayes判别和Fisher判别对目标进行识别。

Fisher判别的基本思想是投影，即将表面上不易分类的数据通过投影到某个方向上，使得投影类与类之间得以分离的一种判别方法。

其判别函数为：

$$W(x) = (\mu_1 - \mu_2)^T \sum^{-1} x - K = (x - \frac{1}{2}(\mu_1 + \mu_2))^T \sum^{-1} (\mu_1 - \mu_2) \quad (4.14)$$

判别规则为：

$$\begin{cases} x \in X_1, W(x) \geq 0 \\ x \in X_2, W(x) < 0 \end{cases} \quad (4.15)$$

Bayes判别和Fisher估计的思想方法是一样的，即假定对研究的对象已经有一定的认识，这种认识常用先验概率来描述，当我们取得一个样本后，就可以用样本来修正已有的先验概率分布，得出后验概率分布，再通过后验概率分布进行各种统计推断。

被判为 X_1 的概率为：

$$P(1|2) = P(x \in R_1 | x_2) = \int_{R_1} \dots \int f_2(x) dx \quad (4.16)$$

模型求解结果如下：

表4_6 无量纲化能量谱特征识别

无量纲化能量谱	“数据包 1”	识别率	
		Bayes 判别	Fisher 判别
Test1ForSence	场景一	93.33%	83.25%
Test2ForSence	场景二	93.33%	82.30%

表4_7 无量纲化半径分布特征识别

无量纲化半径分布	“数据包 1”	识别率	
		Bayes 判别	Fisher 判别
Test1ForSence	场景一	53.33%	94.32%
Test2ForSence	场景二	53.33%	91.56%

由表4_6和表4_7可知，针对无量纲化能量谱，使用Bayes判别准则的识别率高达93.33%，但是对于无量纲化半径分布使用Bayes判别准则的识别率只有53.33%；Fisher判别准则无论是针对哪个特征，都能达到较高的识别率。在这种情况下，使用Fisher判别准则更合适。

(3) 神经网络求解

神经网络是由众多的神经元可调的连接权值连接而成，具有大规模并行处理、分

布式信息存储、良好的自组织自学习能力等特点。BP 神经网络算法在理论上可以逼近任意函数，基本的结构由非线性变化单元组成，具有很强的非线性映射能力。而且网络的中间层数、各层的处理单元数及网络的学习系数等参数可根据具体情况设定，灵活性很大，在优化、信号处理与模式识别、智能控制、故障诊断等许多领域都有着广泛的应用前景。运用神经网络自学习能力的特征，将每个信道数据转置后的 100 各点作为输入，采用 10 各神经元，单输出的结构，如图 4.17 所示。

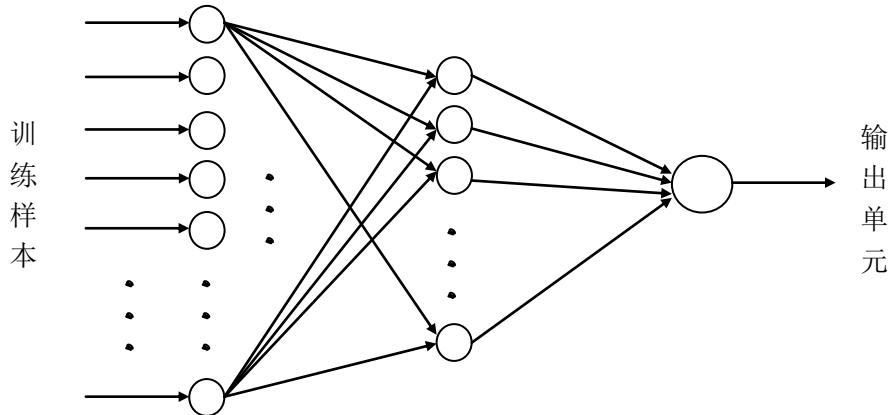


图 4.17 神经网络结构图

以“数据包 1”中的 Test1~Test4 的数据转置后作为训练样本，Test5 作为测试样本，三种场景的网络输出分别为 1、2 和 3，对网络进行自适应学习。

将“数据包2”所给场景作为神经网络的输入，输出与神经网络训练结果1、2和3 做对比来识别场景，网络训练及测试结果如表4_8所示，识别结果如表4_9所示。

表4_8 神经网络训练及测试结果

	训练样本				测试样本	训练输出
场景1	Test1	Test2	Test3	Test4	Test5	——
输出	0.9902	1.0162	0.9972	0.9837	1.3623	1
场景2	Test1	Test2	Test3	Test4	Test5	——
输出	1.9802	1.9942	2.0217	1.9921	1.96403	2
场景3	Test1	Test2	Test3	Test4	Test5	——
输出	3.0176	2.9982	2.9845	2.9911	2.6307	3

表4_9 神经网络识别结果

	神经网络输出结果	识别率
Test1ForScene	1.4647	54.53%
Test2ForScene	1.9848	99.24%

网络自学习后，测试样本的测试结果大体上满足实际要求，该网络可以用来做识别。

(4) 相位和求解

将所给样本所有时段测得数据点的相位（即：信号相角）求和，将所得结果与问题一中场景一、二和三所得数据进行对比，根据接近程度做场景识别。

$$\min = f_{2_i} - f_{1_j}, i=1,2, j=1,2,3 \quad (4.17)$$

判断结果如表4_10所示：

表4_10 识别结果及识别率

	相位和	场景判断	识别率
Test1ForScene	2.3580×10^5	场景一	99.72%
Test2ForScene	2.3525×10^5	场景二	99.91%

通过表4_10的识别结果以及识别率可看出通过相位关系的识别率较其他识别方法是很高。

4.2.3 结果分析

由上述模型对“数据包2”数据进行分类情况可知，问题二中的Test1ForScene数据，采集于问题一中场景一；Test2ForScene数据，采集于问题一中场景二，且均有较高的识别率。采用的四种方案的场景识别率是不同的，通过相位关系对场景的识别率是最高的。这是由于不同场景的多径衰落情况是不同的，各个采样点的相位和反映出了总的多径时延，而场景的多径时延与接收点到发射点的距离无关，只与场景自身特征有关，因此能达到很高的识别率。场景无量纲化后的能量谱以及聚类分析的半径变化率使用Fisher判别准则都有较高的识别率。神经网络和最小二乘法在对Test2ForScene的场景识别率是很高的。

4.3 问题三：对“数据包3”中的路径进行区域划分

4.3.1 问题三分析

对于问题三，由于“数据包3”中数据较多，为降低问题复杂度，采用问题一中建立的统计学分析模型，可先对其数据进行分段，再对统计学分析模型进行改进得到滑动窗口模型并对划分结果进行分析，可得到较合理的“指纹”区域划分。最后利用等高线“指纹”特征可验证划分的“指纹”区域是合理的。

“数据包3”的真实测量信号由前面的相邻信道样本间隔 $2/3\text{ms}$ 变为 10ms ，这对 100 个采样点的相位和不会造成影响。由问题二可知相位分析得到的场景识别率是最高的，针对固定长度接收到的采样点的相位和，能够准确的识别出该接收端所处环境。通过固定窗口长度滑动，来识别滑动过程中窗口覆盖距离数据的采集环境，对给定场景相位特征如表4_11：

表4_11 各场景 1500×100 采样点相位和对应范围

场景	采样点相位和对应范围
场景一	$2.3549 \times 10^5 \sim 2.3607 \times 10^5$
场景二	$2.3494 \times 10^5 \sim 2.3550 \times 10^5$
场景三	$2.3635 \times 10^5 \sim 2.3720 \times 10^5$

根据表4_11，规定：

- (1) 求得相位和小于 2.3494×10^5 ，给定为场景四；
- (2) 求得相位和大于 2.3720×10^5 ，给定场景五；
- (3) 求得相位和介于 2.3607×10^5 与 2.3635×10^5 之间，给定场景零。

为更加有利于模型建立与分析，需对“数据包3”进行数据处理，绘制出如图4.18所示的等高线示意图，从图中可以看出，该图直观反映了样本的指纹特征，可以应用于摘自本段数据的试验验证，同时可以进行对场景的划分和场景变化情况的分析。

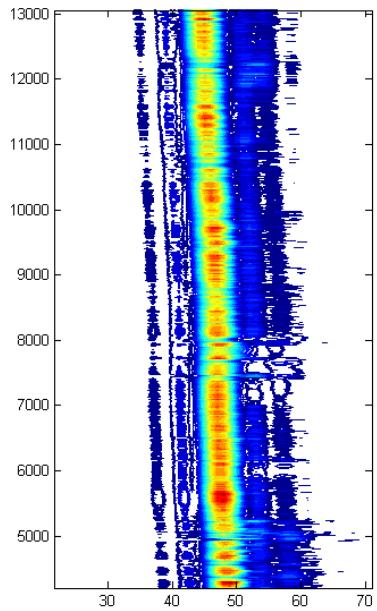


图 4.18 “数据 3” 样本等高线图

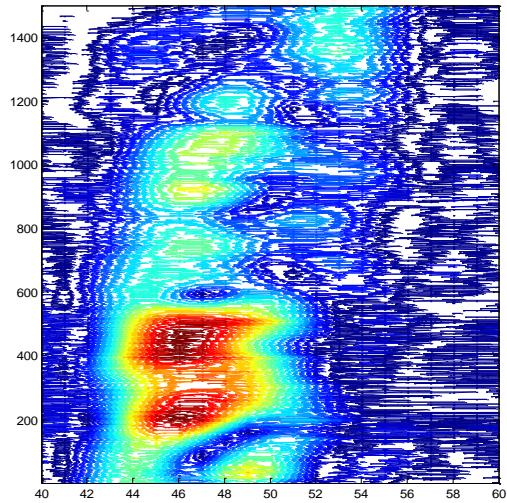


图 4.19 样本 Z 轴等高线图

根据样本等高线图显示，指纹特征十分明显，我们可以将最高点附近的等高线位置和出现的规律作为指纹，并能够较为直观的对场景进行合理的场景划分。由于区域的划分涉及精细与粗糙的均衡，不宜过细或过粗糙，故在可辨别的情况下，我们将场景与距离大致划分为如下表格情形：

表 4_12 区域划分

0	1500	2000	5000	7500	8000	13800	17700
场景一	场景二	场景一	场景三	场景二	场景四	场景二	场景二

4.3.2 模型的建立和求解

对17700*100的测量数据从起始点开始以窗口长度为1500依此向前推进，并依此求

得该窗口相位和，不同范围返回相应的场景。具体模型如下：

$$\theta_z = \sum_{i=1}^{1500} \theta_{i,j}, j=1,2,\dots,100 \quad (4.18)$$

$$\left\{ \begin{array}{l} CJ = 1, \theta_z \in (2.3549e+05, 2.3607e+05) \\ CJ = 2, \theta_z \in (2.3494e+05, 2.3549e+05) \\ CJ = 3, \theta_z \in (2.3635e+05, 2.3720e+05) \\ CJ = 4, \theta_z < 2.3494e+05 \\ CJ = 5, \theta_z > 2.3720e+05 \\ CJ = 0, \theta_z \in (2.3607e+05, 2.3635e+05) \end{array} \right. \quad (4.19)$$

根据上述模型求解得测试路段共有6种场景、340个区域划分，如表4_13所示，其中白色行表示场景的变换点，灰色行小数值0~5表示该点对应场景。由模型可知各场景信道采样点的总的相位和排序为：4<2<1<0<3<5。

表4_13 场景分末点

726	731	737	783	832	842	850	861	872	969
4	2	4	2	1	0	3	0	3	0
975	1003	1212	1224	1226	1227	1230	1377	1864	1865
3	0	3	5	3	5	3	5	3	5
1978	2182	2188	2315	2403	2429	2489	2507	2546	2555
3	5	3	5	3	0	1	0	1	0
2762	2835	2842	2876	2921	2923	2931	2943	3109	3115
1	2	1	2	1	0	1	0	1	0
3123	3165	3166	3167	3177	3215	3221	3231	3239	3240
1	0	1	0	1	0	3	0	1	0
3242	3245	3276	3307	3313	3361	3370	3383	3503	3514
1	0	1	0	3	0	1	0	1	2
3538	3560	3664	3684	4305	4310	4492	4493	4494	4496
1	2	4	2	4	2	4	2	4	2
4497	4498	4500	4504	5194	5202	5203	5244	5327	5338
4	2	4	2	4	2	4	2	1	0
5778	5947	6193	6225	6314	6323	6324	6325	6357	6377
1	2	1	2	1	2	1	2	1	2
6408	6432	6459	6543	6553	6601	6603	6666	6669	6670
1	2	1	2	1	2	1	2	1	2
6671	6805	6806	6812	6815	6816	6817	6829	6841	6854
1	2	4	2	4	2	4	2	4	2
6860	6891	6892	7008	7100	7110	7143	7145	7146	7154
4	2	4	2	4	2	4	2	4	2
7155	7164	7258	7442	7702	7725	7756	7780	7781	7782
4	2	4	2	4	2	4	2	4	2
7783	7784	7797	7970	7978	8069	8072	8076	8077	8115

4	2	4	2	1	2	1	2	1	2
8117	8133	8140	8153	8155	8176	8392	8420	8423	8462
1	2	1	2	1	2	1	2	1	2
8807	8821	9549	9552	9558	9642	9643	9644	9648	9673
1	2	1	2	1	2	1	2	1	2
9674	9891	9898	9935	9955	10132	10478	10482	10511	10512
1	2	1	2	1	2	1	0	1	0
10520	10524	11109	11111	11138	11142	11143	11144	11146	11217
1	0	1	0	1	0	1	0	1	0
11298	11313	11314	11315	11317	11333	11342	11360	11361	11411
3	0	3	0	3	0	3	0	3	0
11412	11413	11414	11505	11515	11546	11553	11564	11577	11578
3	0	3	0	3	0	1	0	1	0
11580	11582	11842	11843	11854	11908	11912	11918	12292	12316
1	0	1	0	1	0	1	0	1	0
12319	12392	12396	12456	12469	12480	12487	12492	12516	12597
3	0	1	0	1	0	1	0	1	0
12616	12626	12630	12638	12642	12676	12792	12843	12845	12861
1	0	1	0	1	0	1	0	3	0
12871	12891	12902	12905	12906	12952	12962	12994	12998	13003
3	0	1	0	1	0	1	0	1	0
13086	13147	13160	13167	13353	13374	13395	13398	13405	13419
1	0	3	0	3	0	1	0	1	0
13429	13461	13533	13550	13554	13568	13666	13669	13698	13704
3	0	1	0	3	0	3	0	3	0
13713	13773	13780	13795	13796	13852	13972	13976	13988	13996
3	0	3	0	1	0	3	0	3	0
14008	14015	14016	14025	14042	14072	14125	14130	14404	14408
3	0	3	0	3	0	1	0	1	0
14451	14461	14468	14477	14486	14495	14509	14580	14588	14649
1	0	1	0	1	0	1	0	3	0
14651	14716	14757	14766	14778	14779	14879	14892	14895	14906
1	0	1	2	1	2	1	0	1	0
14932	14947	14987	14992	15081	15094	15114	15133	15135	15199
1	0	3	0	3	0	1	0	3	0
15202	15242	15245	15257	15276	15282	15283	15284	15299	15333
3	0	3	0	3	0	3	0	3	0
15341	15356	15368	15382	15425	15479	15484	15567	15825	15858
3	0	1	0	1	0	1	0	3	0
15866	15915	15918	15919	15920	15921	16029	16063	16072	16185
3	0	3	0	3	0	3	0	3	0
17700									
0									

由于数据开始时刻区域划分较细, 726~969个信道中场景变换十次, 虽然区域划分越细, 对区域识别的精确程度越高, 但过细的区域划分会在“指纹”特征中引入更多错误。因此, 对表4_12需要按照一定的缩减准则进行缩减, 具体缩减准则如下:

- 1、由于滑动窗的固有长度, 会存在短距离内两个场景连续变化, 根据4-2-1-0-3-5的递增规律, 根据两个场景连续变化起始点相邻场景来统一场景。如: 短时间内出现2-1-0-1-0-1-0-1-0-3规律的场景变换, 可转化为2-1-0-3。
- 2、若在稳定场景内, 出现极短时间的另一场景切换, 将其剔除处理。如: 1(1200)-1(1398)-0(1400)-0(1401)-1(1402)-1(1600), 可将其转变为1(1200)-1(1600)。
- 3、若场景出现短时间的跳跃切换, 对其进行连续处理。如: 1-3变为1-0-3。

根据17700对应150米, 求出场景切换具体位置。按照上面3个准则将表4_13进行缩减得到表4_14。其中, 白色行表示场景的变换点, 灰色行小数值0~5表示该点对应场景, 粉色行表示步行距离。

表4_14 缩减后的分段结果

1	726	790	832	842	850	1224	1978	2429	2489
2	4	2	1	0	3	5	3	0	1
0	6.15	6.69	7.05	7.14	7.20	10.37	16.76	20.58	21.09
2835	2921	3225	3514	5327	6066	7197	7588	9012	10132
2	1	0	2	1	2	4	2	1	0
24.03	24.75	27.33	29.78	45.14	51.41	60.99	64.31	76.37	85.86
11298	11553	12414	12829	13429	13920	14152	14567	14947	15174
3	0	1	0	3	0	1	0	3	0
95. 75	97.91	105.20	108.72	113.81	117.97	119.93	123.45	126.67	128.59
15368	15586	15825	16185	17700					
1	0	3	0	1					
130.24	132.08	134.11	137.16	150					

4.3.3 结果分析

根据前面的分析结果, 使用滑动窗口模型对路段区域划分具有很强的识别性, 区域划分的精度很高, 这是的在“指纹”特征的引入过程引入更多的错误。因此需要对该区域划分进行缩减, 经过缩减后的区域划分更具合理性, 缩减后为6个场景35段.考虑实际情况, 该结果任然需要再缩减。

然而通过等高线自主分段获得7段四个场景(0m, 12.71m, 16.95m, 42.37m, 63.56m, 67.79m, 116.95m, 150m), 等高线的特征极其明显, 因此该结果更具有说服力。

4.4 问题四：对“数据包4”中的样本进行区域识别

4.4.1 问题分析

根据“数据包4”提供的两个真实信道测量结果和问题3中的测量数据, 固定各自窗口为100的数据宽度, 对其采用聚类半径无量纲化后的方差和相位和方差进行特征提取, 通过最小二乘法选取“数据包4”中两个真实测量数据在“数据包3”中最佳匹配位置。为保证“样本判断”的结果真确, 采用经度更高的滑动窗口模型在数据3中求取全局最优解。

4.4.2 模型的建立与求解

将问题三数据和问题四数据分为 $2*100*100$ 和 $177*100*100$ 固定窗口的数据段，针对数据三和数据四分别求聚类半径无量纲化后的方差以及相位和的方差，

聚类半径无量纲化后的方差具体模型如下：

$$\begin{cases} a_0 = \sum_{i=1}^{1500} \sum_{k=1}^{100} a_{ik} / 150000 \\ b_0 = \sum_{i=1}^{1500} \sum_{k=1}^{100} b_{ik} / 150000 \end{cases} \quad (4.20)$$

$$l_{i_k} = [(a_{i_k} - a_0)^2 + (b_{i_k} - b_0)^2]^{1/2}, i = 1, 2, \dots, 100, k = 1, 2, \dots, 100 \quad (4.21)$$

$$\hat{l}_{i_k} = l_{i_k} / [\sum_{i=1}^{1500} l_{i_k} / 100], k = 1, 2, \dots, 100 \quad (4.22)$$

$$\sigma_r = \sum_{i=1}^{1500} (\hat{l}_{i_k} - \hat{l})^2 / 100, k = 1, 2, \dots, 100 \quad (4.23)$$

相位和方差的具体模型如下：

$$\theta_k = \sum_{i=1}^{100} \theta_{k_i}, k = 1, 2, \dots, 100 \quad (4.24)$$

$$\sigma_j = \sum_{i=1}^{100} (\theta_k - \hat{\theta}_k), j = 1, 2, \dots, 177 \quad (4.25)$$

利用前面最小二乘法公式分别求出“数据包4”固定窗口两个连续聚类半径无量纲化方差、相位和方差、“数据包3”177个连续聚类半径无量纲化后的方差和相位和方差，通过最小二乘法找出两者之间的最小距离。即可较为粗略判断“数据包4”提供的两个真实信号是否属于问题三中所提供的路段。

在此基础上改用滑动窗口通过最小二乘法求全局最短距离。模型结果表4_14所示：

表4_14 滑动窗口模型运行结果

方法及误差 样本	聚类半径法 (最佳匹配)	相对距离	相位和方差法 (最佳匹配)	相对距离
SampleForTest1	1800~2000	3.2931e-12	1783~1983	6.7492e-14
SampleForTest2	8900~9100	3.9069e-18	8900~9100	3.9069e-18

由模型结果可知，经过两次寻优SampleForTest2都在“问题3”中8900~9100处的数据，且相对距离均为 $3.9069e-18$ ，几乎可忽略不计，因此可判定SampleForTest2的测量数据是在“问题3”中所提供的路段上采集的；针对SampleForTest1，使用不同方法获得的最佳相对距离不相同且相对SampleForTest2相差4个数量级，因此有理由判定SampleForTest1中的数据不是在“问题3”中的路段中采集的。

4.4.3 模型验证

对样本“数据包3”处理得到的等高线图，我们可以对本问中所得到的结果进行分析验证。同时对模型具有高精度值的情况给予验证。下面是对“数据包4”的样本进行分析，同时对样本 SampleForTest1 和 SampleForTest2 绘制等高线，便于结果的直

观验证，其图像分别如下：

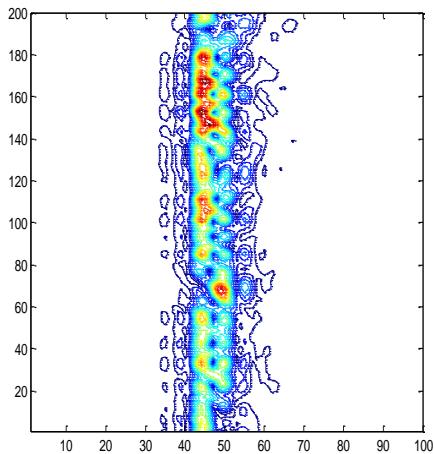


图4.20 SampleForTest1等高线图

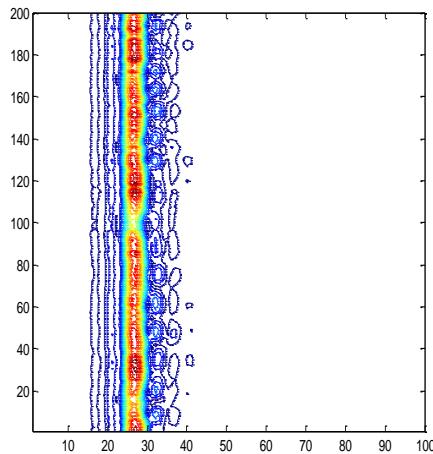


图4.21 SampleForTest2等高线图

4.4.4 聚类半径法结果验证

通过聚类分析，分别得到了样本SampleForTest1和SampleForTest2的最佳匹配位于问题三中数据段的1800~2000行和8900~9100行，现对其进行验证。

如下图3是“数据包3”所给数据处理得到的1800~2000行等高线图，与图1进行对比，他们的指纹特征并不相同，即可验证结果：样本SampleForTest1的数据并非来自“数据包3”。

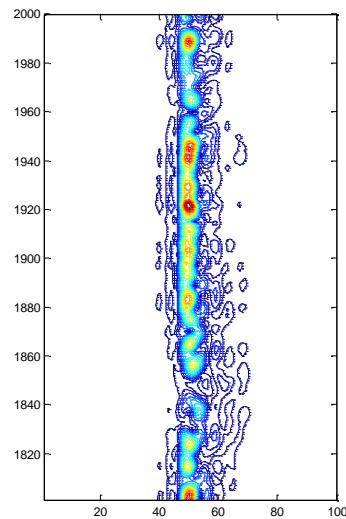


图4.22 “数据包3” 中1800~2000行数据截图

图4是“数据包3”所给数据处理而得到的8800~9200行等高线，图中两横线分别表示“数据包3”中8900~9100行的数据截图，与图4.21进行对比可知，样本SampleForTest2的等高线截图与图4.22完全一致，即证得该样本摘自“数据包3”所给区域。

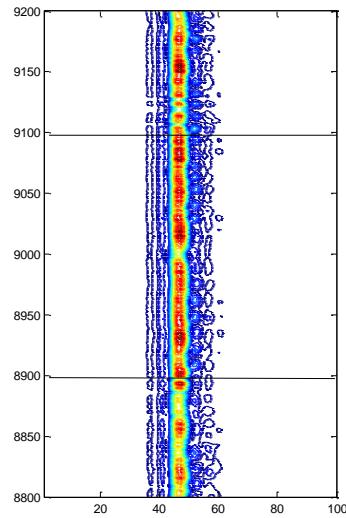


图4.23 SampleForTest2对应在原始图中的位置

4.4.5 相位和方差法验证

如下图4.24是从样本“数据包3”所给数据处理而得到的4200~4400行等高线图。截取4292行以上的数据，同理，与图1进行对比，可知他们的指纹特征并不相同，即可验证结果：样本SampleForTest2中的数据并非来自“数据包3”所给区域。

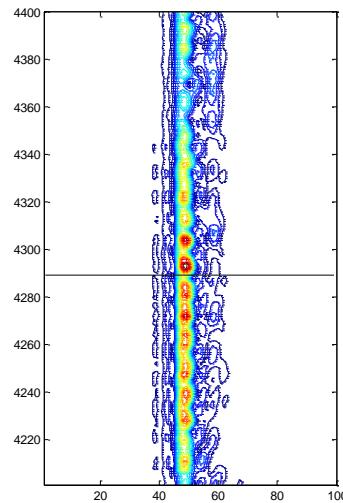


图4.24 “数据包3”中4200~4400行数据截图

相位和方差法处理SampleForTest2所得到的最佳匹配位置同样位于“数据包3”中8900~9100行的数据，由前已证得该段数据与“数据包3”8900~9100行的数据完全匹配，故可知此方法其结果也具有可靠性。

五、模型的评价与扩展

5.1 模型的评价

5.1.1 模型的优点

(1) 问题一中建立的模型是基于统计分析和聚类分析完成，从多方面各角度进行了无线信道“指纹”特征提取，包括各信道相对衰落参数、相对时延、半径归一化后的变化率以及功率归一化后的方差特征，并从2维至3维的角度观察图形特征，简单快速，容易理解。

(2) 问题二中利用模型一的方法求出各信道数据的统计分析特征、归一化半径变化率以及归一化功率方差，通过最小二乘法、判别分析（贝叶斯准则和fisher准则）对各场景进行识别，通过计算，每个方法的误判率都比较低。根据问题二的特性，同样也可以用人工神经网络进行解答，但是需要问题一的数据包作为训练数据，设置合理参数，训练完毕后，可直接进行判断。

(3) 问题三中，分析了等高线特征图，其特征清晰明了，易于观察。

(4) 问题四中所建立的基于聚类分析半径应用窗口滑动模型可精确得到数据包四的样本数据在数据包三的位置，并可精确地用等高线特征图进行识别并验证。

5.1.2 模型的缺点

(1) 通过问题一中的算法求解出来的参数和图像，可能存在一定的误差，图像中的坐标存在一定误差，需要进一步讨论，减小这种误差。

(2) 在问题二的场景识别中，可能由于提供的数据量相对较少所导致人工神经网络训练样本不足，进而导致识别率不是很高。

5.2 模型的扩展

5.2.1 模型改进

(1) 在模型建立中考虑数据去噪声和信道数据还原。信号在发送端和接收端都有用滤波器进行滤波，且在信号传输过程中不可避免地伴随着高斯白噪声，去噪后，我们得到的各种参数将会更精准。

(2) 可在此研究基础上，进一步分析和提取图像特征，建立空间图像特征的函数特征，可针对不同的场合进行场景切换的自适应识别。

5.2.2 模型推广

我们通过场景特征提取模型、场景识别模型可以对不同的场景进行提取特征和识别，进而我们在进行无线通信时可以根据不同的场景情况对通信信号的不同处理来避免信号的失真。同时，通过区域识别模型，我们能使误判距离尽可能小，在高速运行的接收端，也能有很好的通信效果。

六、参考文献

- [1] 万喆. 若干种盲信道辨识与均衡算法的研究, [硕士学位论文]. 西安:西安电子科技大学, 2003.
- [2] 吴启军. 电台指纹识别算法研究, [硕士学位论文]. 西安:西安电子科技大学, 2010.
- [3] 程文璞. 高速移动环境中基于理论方法的无线信道建模研究, [硕士学位论文]. 北京:北京邮电大学, 2014.
- [4] 庄铭杰, 郭东辉. 移动通信中无线信道特性的研究[J]. 电讯技术, 2004, (5):34-40.
- [5] 宋家琪. 基于邻域粗糙集的信号指纹属性简约, [硕士学位论文]. 北京:北京邮电大学, 2011.
- [6] 王莅康. 基于信道特征的协作密钥提取技术研究, [硕士学位论文]. 上海:上海交通大学, 2011.
- [7] 张贤达, 保铮著, 通信信号处理. 北京: 国防工业出版社, 2000
- [8] 万永革编著, 数字信号处理的 MATLAB 实现. 南京: 科学出版社, 2006
- [9] 盛骤, 谢式千, 潘承毅等, 概率论与数理统计. 北京: 高等教育出版社, 2011
- [10] 冯松. 无线信道测量参数提取算法研究, [硕士学位论文]. 西安: 西安电子科技大学, 2013.
- [11] 赵静, 但琦. 数学模型[M]. 3 版. 北京: 高等教育出版社, 2003.
- [12] 李建东, 郭梯云, 邬国扬著, 移动通信. 山西: 西安电子科技大学出版社, 2012

七、附 件

附录:

问题一源程序: power_1.m、angle1.m、distance_std_changerate1.m、ann.m、jian_ce.m
power_1.m 表示运用功率归一化方差建立指纹特征图像模型
angle1.m 表示运用相位和归一化方差建立指纹特征模型
distance_std_changerate1.m 表示运用聚类分析建立聚类半径和变化率的指纹模型
ann.m 神经网络训练文件
jian_ce 神经网络检测文件

问题二源程序: angle_judge.m 、fisher.m、 powerstd_judge.m、 radiusstd_judge.m
angle_judge.m 表示相位和归一化方差判别分类
fisher.m 表示 fisher 准则判别分析函数
powerstd_judge.m 表示功率归一化方差判别分类
radiusstd_judge.m 表示用聚类半径进行判别分类

问题三源程序: angle_divide.m、julei_divide.m
angle_divide.m 表示运用相位指纹特征模型来进行场景划分
julei_divide.m 表示运用菌类半径指纹特征模型来进行场景划分

问题四源程序: angle_divide1.m、dist_std_divide.m、picture1_divde、sameple1_divde.m、angle_smoothwindow、julei_smoothwindow.m

angle_divide1.m 表示运用相位进行样本匹配

dist_std_divide.m 表示运用聚类半径方差进行样本匹配以及滑动窗口进行距离优化

picture1_divde 表示运用图像特征模型进行样本匹配

angle_smoothwindow 表示运用相位的单位滑动窗口匹配

julei_smoothwindow.m 表示运用聚类半径的单位滑动窗口匹配

图形绘制源程序: kongjian_tu.m、kongjian_tu_new.m、kongjian_tu_jianche_2.m、changjing_sample.m、changjing_1.m 等

部分源程序如下:

```
% -----power_1.m-----%
```

```
clear all
```

```
clc
```

```
load Test1.mat
```

```
t1=ChannellR;
```

```
r=real(t1);
```

```
i=imag(t1);
```

```
X=r.^2+i.^2;
```

```
X_max=max(max(X));
```

```
for j=1:1500
```

```
    power(j)=sum(X(j,:))/100;
```

```
end
```

```
max_power=max(power);
```

```
power_means=power/max_power;
```

```
X=X/X_max;
```

```
power_std=std2(X)
```

```
figure(1)
```

```
plot(power_means,'-')
```

```
hold on
```

```
load Test2.mat
```

```
t2=ChannellR;
```

```
r=real(t2);
```

```
i=imag(t2);
```

```
X=r.^2+i.^2;
```

```
X_max=max(max(X));
```

```
for j=1:1500
```

```
    power1(j)=sum(X(j,:))/100;
```

```
end
```

```
max_power1=max(power1);
```

```
power_means1=power/max_power1;
```

```
X=X/X_max;
```

```

power_std1=std2(X)
plot(power_means1,'-k')

load Test3.mat
t3=ChannelIR;
r=real(t3);
i=imag(t3);
X=r.^2+i.^2;
X_max=max(max(X));
for j=1:1500
    power2(j)=sum(X(j,:))/100;
end
max_power2=max(power2);
power_means2=power/max_power2;
X=X/X_max;
power_std2=std2(X)
plot(power_means2,'-r')

load Test4.mat
t4=ChannelIR;
r=real(t4);
i=imag(t4);
X=r.^2+i.^2;
X_max=max(max(X));
for j=1:1500
    power3(j)=sum(X(j,:))/100;
end
max_power3=max(power3);
power_means3=power/max_power3;
X=X/X_max;
power_std3=std2(X)
plot(power_means3,'-m')

load Test5.mat
t5=ChannelIR;
r=real(t5);
i=imag(t5);
X=r.^2+i.^2;
X_max=max(max(X));
for j=1:1500
    power4(j)=sum(X(j,:))/100;
end
max_power4=max(power4);
power_means4=power/max_power4;

```

```

X=X/X_max;
power_std4=std2(X)
plot(power_means4,'-c')
xlabel('信道数')
ylabel('归一化功率 (无量纲) ')
legend('Test1','Test2','Test3','Test4','Test5')
%-----angel_1.m-----%
clear all
clc
load Test1.mat
t1=ChannelIR;
sum1=sum(sum(abs(angle(t1))));
load Test2.mat
t2=ChannelIR;
sum2=sum(sum(abs(angle(t2))));
load Test3.mat
t3=ChannelIR;
sum3=sum(sum(abs(angle(t3))));
load Test4.mat
t4=ChannelIR;
sum4=sum(sum(abs(angle(t4))));
load Test5.mat
t5=ChannelIR;
sum5=sum(sum(abs(angle(t5))));
sum_all=[sum1 sum2 sum3 sum4 sum5]
sum_std=std(sum_all)
%----- distance_std_changerate1.m-----%
clear all
clc
load Test1.mat
t1=ChannelIR;
r=real(t1);
r_means=sum(sum(r))/150000;
i=imag(t1);
i_means=sum(sum(i))/150000;
X=sqrt(r.^2+i.^2);
%欧式距离计算
dist_oushi=sqrt((r-r_means).^2+(i-i_means).^2);
for i=1:100
    dist(i)=sum(dist_oushi(:,i))/1500;
end
dist_std=std(dist)
dist_max=max(dist);
dist_means=dist/dist_max;

```

```

dist_change_rate=diff(dist_means);
figure(1)
plot(dist_means,'-*');
hold on

load Test2.mat
t2=ChannellR;
r1=real(t2);
r1_means=sum(sum(r1))/150000;
i1=imag(t2);
i1_means=sum(sum(i1))/150000;
X1=sqrt(r1.^2+i1.^2);
% 欧式距离计算
dist1_oushi=sqrt((r1-r1_means).^2+(i1-i1_means).^2);
for i=1:100
    dist1(i)=sum(dist1_oushi(:,i))/1500;
end
dist1_std=std(dist1)
dist1_max=max(dist1);
dist1_means=dist1/dist1_max;
dist1_change_rate=diff(dist1_means);
plot(dist1_means,'-ok');

load Test3.mat
t3=ChannellR;
r2=real(t3);
r2_means=sum(sum(r2))/150000;
i2=imag(t3);
i2_means=sum(sum(i2))/150000;
% 欧式距离计算
dist2_oushi=sqrt((r2-r2_means).^2+(i2-i2_means).^2);
for i=1:100
    dist2(i)=sum(dist2_oushi(:,i))/1500;
end
dist2_std=std(dist2)
dist2_max=max(dist2);
dist2_means=dist2/dist2_max;
dist2_change_rate=diff(dist2_means);
% figure(3)
plot(dist2_means,'-xr');

load Test4.mat
t4=ChannellR;
r3=real(t4);

```

```

r3_means=sum(sum(r3))/150000;
i3=imag(t4);
i3_means=sum(sum(i3))/150000;
%欧式距离计算
dist3_oushi=sqrt((r3-r3_means).^2+(i3-i3_means).^2);
for i=1:100
    dist3(i)=sum(dist3_oushi(:,i))/1500;
end
dist3_std=std(dist3)
dist3_max=max(dist3);
dist3_means=dist3/dist3_max;
dist3_change_rate=diff(dist3_means);
plot(dist3_means,'-sm');

load Test5.mat
t5=ChannellIR;
r4=real(t5);
i4=imag(t5);
r4_means=sum(sum(r4))/150000;
i4_means=sum(sum(i4))/150000;
%欧式距离计算
dist4_oushi=sqrt((r4-r4_means).^2+(i4-i4_means).^2);
for i=1:100
    dist4(i)=sum(dist4_oushi(:,i))/1500;
end
dist4_std=std(dist4)
dist4_max=max(dist4);
dist4_means=dist4/dist4_max;
dist4_change_rate=diff(dist4_means);
plot(dist4_means,'-dc');
xlabel('类别数')
ylabel('归一化半径 (无量纲) ')
legend('Test1','Test2','Test3','Test4','Test5')
grid on
hold off
dist_all=[dist_std dist1_std dist2_std dist3_std dist4_std];
dist_all_std=std(dist_all)
%半径变化率
figure(2)
plot(dist_change_rate,'-*');
grid on
hold on
plot(dist1_change_rate,'-ok');
plot(dist2_change_rate,'-xr');

```

```

plot(dist3_change_rate,'-sm');
plot(dist4_change_rate,'-dc');
xlabel('类别数')
ylabel('归一化半径变化率 (无量纲) ')
legend('Test1','Test2','Test3','Test4','Test5')

%----- ann.m -----
clear all
clc
load Test15.mat
t11=ChannelIR;
t11_mod=max(max(abs(t11)));
t11=t11/t11_mod;
load Test12.mat
t12=ChannelIR;
t12_mod=max(max(abs(t12)));
t12=t12/t12_mod;
load Test13.mat
t13=ChannelIR;
t13_mod=max(max(abs(t13)));
t13=t13/t13_mod;

load Test14.mat
t14=ChannelIR;
t14_mod=max(max(abs(t14)));
t14=t14/t14_mod;

load Test25.mat
t21=ChannelIR;
t21_mod=max(max(abs(t21)));
t21=t21/t21_mod;

load Test22.mat
t22=ChannelIR;
t22_mod=max(max(abs(t22)));
t22=t22/t22_mod;

load Test23.mat
t23=ChannelIR;
t23_mod=max(max(abs(t23)));
t23=t23/t23_mod;

load Test24.mat

```

```

t24=ChannelIR;
t24_mod=max(max(abs(t24)));
t24=t24/t24_mod;

load Test35.mat
t31=ChannelIR;
t31_mod=max(max(abs(t31)));
t31=t31/t31_mod;

load Test32.mat
t32=ChannelIR;
t32_mod=max(max(abs(t32)));
t32=t32/t32_mod;

load Test33.mat
t33=ChannelIR;
t33_mod=max(max(abs(t33)));
t33=t33/t33_mod;

load Test34.mat
t34=ChannelIR;
t34_mod=max(max(abs(t34)));
t34=t34/t34_mod;

t=[t11;t12;t13;t14;t21;t22;t23;t24;t31;t32;t33;t34];

total_t=t';

P=total_t;
T=ones(1,18000);
for i=6001:12000
    T(i)=2;
end
for i=12001:18000
    T(i)=3;
end
p=abs(P);
Pr=ones(100,2);
for i=1:100
    Pr(i,1)=0;
end

net=newff(Pr,[10 1],{'tansig','purelin'},'traincgf');

```

```

net.trainParam.epochs=500;
net.trainParam.goal=0.001;
net.trainParam.show=1;
net.trainParam.lr=0.05;
net =train(net,p,T);

%-----jian_ce.m-----
load Test11.mat
t15_1=ChannellIR;
t15_1=abs(t15_1)';
t15_1=t15_1/max(max(t15_1));
s=sim(net,t15_1);
sum(s)/1500

load Test12.mat
t15_1=ChannellIR;
t15_1=abs(t15_1)';
t15_1=t15_1/max(max(t15_1));
s=sim(net,t15_1);
sum(s)/1500

load Test13.mat
t15_1=ChannellIR;
t15_1=abs(t15_1)';
t15_1=t15_1/max(max(t15_1));
s=sim(net,t15_1);
sum(s)/1500

load Test14.mat
t15_1=ChannellIR;
t15_1=abs(t15_1)';
t15_1=t15_1/max(max(t15_1));
s=sim(net,t15_1);
sum(s)/1500

load Test15.mat
t15_1=ChannellIR;
t15_1=abs(t15_1)';
t15_1=t15_1/max(max(t15_1));
s=sim(net,t15_1);
sum(s)/1500

load Test21.mat
t15_1=ChannellIR;

```

```
t15_1=abs(t15_1)';
t15_1=t15_1/max(max(t15_1));
s=sim(net,t15_1);
sum(s)/1500
```

```
load Test22.mat
t15_1=ChannellR;
t15_1=abs(t15_1)';
t15_1=t15_1/max(max(t15_1));
s=sim(net,t15_1);
sum(s)/1500
```

```
load Test23.mat
t15_1=ChannellR;
t15_1=abs(t15_1)';
t15_1=t15_1/max(max(t15_1));
s=sim(net,t15_1);
sum(s)/1500
load Test24.mat
t15_1=ChannellR;
t15_1=abs(t15_1)';
t15_1=t15_1/max(max(t15_1));
s=sim(net,t15_1);
sum(s)/1500
```

```
load Test25.mat
t15_1=ChannellR;
t15_1=abs(t15_1)';
t15_1=t15_1/max(max(t15_1));
s=sim(net,t15_1);
sum(s)/1500
```

```
load Test31.mat
t15_1=ChannellR;
t15_1=abs(t15_1)';
t15_1=t15_1/max(max(t15_1));
s=sim(net,t15_1);
sum(s)/1500
```

```
load Test32.mat
t15_1=ChannellR;
t15_1=abs(t15_1)';
t15_1=t15_1/max(max(t15_1));
s=sim(net,t15_1);
```

```
sum(s)/1500
```

```
load Test33.mat  
t15_1=ChannelIR;  
t15_1=abs(t15_1)';  
t15_1=t15_1/max(max(t15_1));  
s=sim(net,t15_1);  
sum(s)/1500
```

```
load Test34.mat  
t15_1=ChannelIR;  
t15_1=abs(t15_1)';  
t15_1=t15_1/max(max(t15_1));  
s=sim(net,t15_1);  
sum(s)/1500
```

```
load Test35.mat  
t15_1=ChannelIR;  
t15_1=abs(t15_1)';  
t15_1=t15_1/max(max(t15_1));  
s=sim(net,t15_1);  
sum(s)/1500
```

```
sum(sim(net,abs(P(:,1:6000)))/6000  
sum(sim(net,abs(P(:,6001:12000)))/6000  
sum(sim(net,abs(P(:,12001:18000)))/6000
```

```
sum(sim(net,abs(P(:,1:1500)))/1500  
sum(sim(net,abs(P(:,1501:3000)))/1500  
sum(sim(net,abs(P(:,3001:4500)))/1500  
sum(sim(net,abs(P(:,4501:6000)))/1500
```

```
sum(sim(net,abs(P(:,6001:7500)))/1500  
sum(sim(net,abs(P(:,7501:9000)))/1500  
sum(sim(net,abs(P(:,9001:10500)))/1500  
sum(sim(net,abs(P(:,10501:12000)))/1500
```

```
sum(sim(net,abs(P(:,12001:13500)))/1500  
sum(sim(net,abs(P(:,13501:15000)))/1500  
sum(sim(net,abs(P(:,15001:16500)))/1500  
sum(sim(net,abs(P(:,16501:18000)))/1500
```

```
% 数据包二数据分类场景
```

```

load Test1ForScene.mat
t15_1=ChannelIR;
t15_1=abs(t15_1)';
t15_1=t15_1/max(max(t15_1));
s=sim(net,t15_1);
sum(s)/1500

load Test2ForScene.mat
t15_1=ChannelIR;
t15_1=abs(t15_1)';
t15_1=t15_1/max(max(t15_1));
s=sim(net,t15_1);
sum(s)/1500

```

问题二部分源程序：

```

% ----- angle_judge.m-----%
clear all
clc
load Test1ForScene.mat
t1=ChannelIR;
sum1=sum(sum(abs(angle(t1([1:300],:))))));
sum2=sum(sum(abs(angle(t1([301:600],:))))));
sum3=sum(sum(abs(angle(t1([601:900],:))))));
sum4=sum(sum(abs(angle(t1([901:1200],:))))));
sum5=sum(sum(abs(angle(t1([1201:1500],:))))));
sum_all1=[sum1 sum2 sum3 sum4 sum5];
sum_all0=sum(sum_all1)
sum_std1=std(sum_all1)

load Test2ForScene.mat
t2=ChannelIR;
sum1=sum(sum(abs(angle(t2([1:300],:))))));
sum2=sum(sum(abs(angle(t2([301:600],:))))));
sum3=sum(sum(abs(angle(t2([601:900],:))))));
sum4=sum(sum(abs(angle(t2([901:1200],:))))));
sum5=sum(sum(abs(angle(t2([1201:1500],:))))));
sum_all2=[sum1 sum2 sum3 sum4 sum5];
sum_all0=sum(sum_all2)
sum_std2=std(sum_all2)

% 模型三：角度模型判别情形
% 1 判别分析法判别场景(fisher 准则法)
disp('判别分析法 (fisher 准则) 判别场景:')
a=[2.3585e+05;2.3562e+05;2.3607e+05;2.3549e+05;2.3582e+05]

```

```

2.3509e+05;2.3536e+05;2.3590e+05;2.3494e+05;2.3550e+05
2.3668e+05;2.3663e+05;2.3846e+05;2.3720e+05;2.3635e+05];
x1=[2.3580e+05;2.3525e+05];
g=[1*ones(5,1);2*ones(5,1);3*ones(5,1)];
[class,err]=fisher(x1,a,g)
%2 判别分析法判别场景 (贝叶斯准则)
disp('判别分析法 (贝叶斯准则) 判别场景:')
[class,err]=classify(x1,a,g)
%3 最小二乘法判别场景
disp('最小二乘法判别场景')
judge1(1)=sqrt(sum((x1(1,1)-a([1:5],1)).^2));
judge1(2)=sqrt(sum((x1(1,1)-a([6:10],1)).^2));
judge1(3)=sqrt(sum((x1(1,1)-a([11:15],1)).^2));
[E1 q1]=sort(judge1);
class=q1(1)
err=E1(q1(1))
judge2(1)=sqrt(sum((x1(2,1)-a([1:5],1)).^2));
judge2(2)=sqrt(sum((x1(2,1)-a([6:10],1)).^2));
judge2(3)=sqrt(sum((x1(2,1)-a([11:15],1)).^2));
[E2 q2]=sort(judge2);
class=q2(1)
err=E2(q2(1))

%----- fisher.m-----%
function [outclass,TabCan,TabL,TabCon,TabM,TabG,trainscore] =
fisher(sampledata,training,group,contri)
%FISHER 判别分析.
if nargin < 3
    error('错误： 输入参数太少， 至少需要 3 个输入。');
end

% 根据分组变量生成索引向量 gindex, 组名元胞向量 groups, 组水平向量 glevels
[gindex,groups,glevels] = grp2idx(group);
% 忽略缺失数据
nans = find(isnan(gindex));
if ~isempty(nans)
    training(nans,:)= [];
    gindex(nans)= [];
end
ngroups = length(groups);
gsize = hist(gindex,1:ngroups);
nonemptygroups = find(gsize>0);
nusedgroups = length(nonemptygroups);

```

```

% 判断是否有空的组
if ngroups > nusedgroups
    warning('警告: 有空的组.');
end

[n,d] = size(training);
if size(gindex,1) ~= n
    error('错误: 输入参数大小不匹配, GROUP 与 TRAINING 必须具有相同的行数。');
elseif isempty(sampleddata)
    sampleddata = zeros(0,d,class(sampleddata));
elseif size(sampleddata,2) ~= d
    error('错误: 输入参数大小不匹配, SAMPLEDATA 与 TRAINING 必须具有相同的列数。');
end

% 设置 contri 的默认值为 1, 并限定 contri 在(0, 1]内取值
if nargin < 4 || isempty(contri)
    contri = 1;
end
if ~isscalar(contri) || contri > 1 || contri <= 0
    error('错误: contri 必须是一个在(0, 1]内取值的标量。');
end

if any(gsize == 1)
    error('错误: TRAINING 中的每个组至少应有两个观测。');
end

% 计算各组的组均值
gmeans = NaN(ngroups, d);
for k = nonemptygroups
    gmeans(k,:) = mean(training(gindex==k,:),1);
end
% 计算总均值
totalmean = mean(training,1);

% 计算组内离差平方和矩阵 E 和组间离差平方和矩阵 B
E = zeros(d);
B = E;
for k = nonemptygroups
    % 分别估计各组的组内离差平方和矩阵。
    [Q,Rk] = qr(bsxfun(@minus,training(gindex==k,:),gmeans(k,:)), 0);
    % 各组的组内离差平方和矩阵: AkHat = Rk'*Rk
    % 判断各组的组内离差平方和矩阵的正定性
    s = svd(Rk);
    if any(s <= max(gsize(k),d) * eps(max(s)))

```

```

error('错误: TRAINING 中各组的组内离差平方和矩阵必须是正定矩阵.');
end

E = E + Rk'*Rk;      % 计算总的组内离差平方和矩阵 E
% 计算组间离差平方和矩阵 B
B = B + (gmeans(k,:)-totalmean)'*(gmeans(k,:)-totalmean)*gsize(k);
end

% 求 inv(E)*B 的正特征值与相应的特征向量
EB = E\B;
[V, D] = eig(EB);
D = diag(D);
[D, idD] = sort(D,'descend'); % 将特征值按降序排列
V = V(:,idD);
NumPosi = min(nGroups-1, d); % 确定正特征值个数
D = D(1:NumPosi, :);
CumCont = cumsum(D/sum(D)); % 计算累积贡献率

% 以表格形式返回所有特征值, 贡献率, 累积贡献率等. TabL 是一个元胞数组
head = {'Eigenvalue', 'Difference', 'Proportion', 'Cumulative'};
TabL = cell(NumPosi+1, 4);
TabL(1,:) = head;
TabL(2:end,1) = num2cell(D);
if NumPosi == 1
    TabL(2:end-1,2) = {0};
else
    TabL(2:end-1,2) = num2cell(-diff(D));
end
TabL(2:end,3) = num2cell(D/sum(D));
TabL(2:end,4) = num2cell(CumCont);

% 根据累积贡献率的下限 contri 确定需要使用的判别式个数 CumContGeCon
CumContGeCon = find(CumCont >= contri);
CumContGeCon = CumContGeCon(1);
V = V(:, 1:CumContGeCon); % 需要使用的判别式系数矩阵

% 以表格形式返回所用判别式的系数向量, 若 contri 取值为 1,
% 则返回所有判别式的系数向量. TabCan 是一个元胞数组
TabCan = cell(d+1, CumContGeCon+1);
TabCan(1, 1) = {'Variable'};
TabCan(2:end, 1) = strcat('x', cellstr(num2str((1:d))));
TabCan(1, 2:end) = strcat('can', cellstr(num2str((1:CumContGeCon))'));
TabCan(2:end, 2:end) = num2cell(V);

% 将训练样品与待判样品放在一起进行判别

```

```

m = size(sampledata,1);
gv = gmeans*V;
stv = [sampledata; training]*V;
nstv = size(stv, 1);
message = "";
outclass = NaN(nstv, 1);
for i = 1:nstv
    obji = bsxfun(@minus,stv(i,:),gv);
    obji = sum(obji.^2, 2);
    idclass = find(obji == min(obji));
    if length(idclass) > 1
        idclass = idclass(1);
        message = '警告：出现了一个或多个结';
    end
    outclass(i) = idclass;
end
warning(message);
triclass = outclass(m+(1:n)); % 训练样品的判别结果（由类序号构成的向量）
outclass = outclass(1:m); % 待判样品的判别结果（由类序号构成的向量）
outclass = glevels(outclass,:); % 将待判样品的判别结果进行一个类型转换

trg1 = groups(gindex); % 训练样品的初始类名称
trg2 = groups(triclass); % 训练样品经判别后的类名称
% 以表格形式返回混淆矩阵（包含总的分类信息的矩阵）. TabCon 是一个元胞数组
[CLMat, order] = confusionmat(trg1,trg2);
TabCon = {[{'From/To'},order];order, num2cell(CLMat)};

% 以表格形式返回误判矩阵. TabM 是一个元胞数组
miss = find(gindex ~= triclass); % 训练样品中误判样品的编号
head1 = {'Obj', 'From', 'To'};
TabM = [head1; num2cell(miss), trg1(miss), trg2(miss)];

% 将所用判别式作用在各组的组均值上，得到组均值投影矩阵，以表格形式返回这个矩阵。
% TabG 是一个元胞数组
TabG = cell/ngroups+1,CumContGeCon+1);
TabG(:,1) = {[{'Group'}];groups};
TabG(1,2:end) = strcat('can',cellstr(num2str((1:CumContGeCon))));
TabG(2:end,2:end) = num2cell(gv);

% 计算训练样品所对应的判别式得分
trainscore = training*V;
trainscore = [gindex, trainscore];
%----- powerstd_judge.m-----%
clear all

```

```

clc
load Test1ForScene.mat
t1=ChannelIR;
r=real(t1);
i=imag(t1);
X1=r.^2+i.^2;
X1_max=max(max(X1));
for j=1:1500
    power(j)=sum(X1(j,:))/100;
end
max_power=max(power);
power_means=power/max_power;
X1=X1/X1_max;
disp('样本 1 的归一化功率方差');
power_std=std2(X1)
figure(1)
plot(power_means,'-')
xlabel('信道数')
ylabel('归一化功率 (无量纲)')
title('样本一归一化功率谱')

load Test2ForScene.mat
t2=ChannelIR;
r=real(t2);
i=imag(t2);
X2=r.^2+i.^2;
X2_max=max(max(X2));
for j=1:1500
    power1(j)=sum(X2(j,:))/100;
end
max_power1=max(power1);
power_means1=power1/max_power1;
X2=X2/X2_max;
disp('样本 2 的归一化功率方差');
power_std1=std2(X2)
figure(2)
plot(power_means1,'-')
xlabel('信道数')
ylabel('归一化功率 (无量纲)')
title('样本二归一化功率谱')

% 模型一：归一化功率模型判别情形
% 1 判别分析法判别场景(fisher 准则法)
disp('判别分析法 (fisher 准则) 判别场景:')

```

```

a=[0.0462;0.0659;0.0592;0.0700;0.0701
    0.0510; 0.0762;0.0822;0.0644;0.0965
    0.0891;0.0687;0.0541;0.0788;0.0740];
x1=[0.0568;0.0844];
g=[1*ones(5,1);2*ones(5,1);3*ones(5,1)];
[class,err]=fisher(x1,a,g)
%2 判别分析法判别场景（贝叶斯准则）
disp('判别分析法（贝叶斯准则）判别场景：')
[class,err]=classify(x1,a,g)
%3 最小二乘法判别场景
disp('最小二乘法判别场景：')
judge1(1)=sqrt(sum((x1(1,1)-a([1:5],1)).^2));
judge1(2)=sqrt(sum((x1(1,1)-a([6:10],1)).^2));
judge1(3)=sqrt(sum((x1(1,1)-a([11:15],1)).^2));
[E1 q1]=sort(judge1);
class=q1(1)
err=E1(q1(1))
judge2(1)=sqrt(sum((x1(2,1)-a([1:5],1)).^2));
judge2(2)=sqrt(sum((x1(2,1)-a([6:10],1)).^2));
judge2(3)=sqrt(sum((x1(2,1)-a([11:15],1)).^2));
[E2 q2]=sort(judge2);
class=q2(1)
err=E2(q2(1))
%----- radiusstd_judge.m-----%
clear all
clc
load Test1ForScene.mat
t1=ChannelIR;
r1=real(t1);
r1_means=sum(sum(r1))/150000;
i1=imag(t1);
i1_means=sum(sum(i1))/150000;
%欧式距离计算
dist1_oushi=sqrt((r1-r1_means).^2+(i1-i1_means).^2);
for i=1:100
    dist1(i)=sum(dist1_oushi(:,i))/1500;
end
dist1_std=std(dist1)
dist1_max=max(dist1);
dist1_means=dist1/dist1_max;
dist1_change_rate=diff(dist1_means);
plot(dist1_means,'-dc');
xlabel('类别数')
ylabel('归一化半径（无量纲）')

```

```

legend('Test1ForScene','Test2ForScene')
title('测试数据归一化半径')
grid on
hold on
load Test2ForScene.mat
t2=ChannelIR;
r2=real(t2);
r2_means=sum(sum(r2))/150000;
i2=imag(t2);
i2_means=sum(sum(i2))/150000;
% 欧式距离计算
dist2_oushi=sqrt((r2-r2_means).^2+(i2-i2_means).^2);
for i=1:100
    dist2(i)=sum(dist2_oushi(:,i))/1500;
end
dist2_std=std(dist2)
dist2_max=max(dist2);
dist2_means=dist2/dist2_max;
dist2_change_rate=diff(dist2_means);
plot(dist2_means,'-*m');
% 半径变化率
% figure(2)
% plot(dist_change_rate,'-*');
% grid on
% hold on
% plot(dist1_change_rate,'-ok');
% plot(dist2_change_rate,'-xr');
% plot(dist3_change_rate,'-sm');
% plot(dist4_change_rate,'-dc');
% xlabel('类别数')
% ylabel('归一化半径变化率 (无量纲) ')
% legend('Test1','Test2','Test3','Test4','Test5')

% 模型二：半径归一化方差模型判别情形
% 1 判别分析法判别场景(fisher 准则法)
disp('判别分析法 (fisher 准则) 判别场景:')
a=[0.3462;0.3807;0.3221;0.7696;0.6898
    0.1128; 0.1258;0.1497;0.2021;0.0448
    0.0477;0.0272;0.0499;0.0624;0.0595];
x1=[0.8325;0.1770];
g=[1*ones(5,1);2*ones(5,1);3*ones(5,1)];
[class,err]=fisher(x1,a,g)
% 2 判别分析法判别场景 (贝叶斯准则)
disp('判别分析法 (贝叶斯准则) 判别场景:')

```

```

[class,err]=classify(x1,a,g)
%3 最小二乘法判别场景
disp('最小二乘法判别场景:')
judge1(1)=sqrt(sum((x1(1,1)-a([1:5],1)).^2));
judge1(2)=sqrt(sum((x1(1,1)-a([6:10],1)).^2));
judge1(3)=sqrt(sum((x1(1,1)-a([11:15],1)).^2));
[E1 q1]=sort(judge1);
class=q1(1)
err=E1(q1(1))
judge2(1)=sqrt(sum((x1(2,1)-a([1:5],1)).^2));
judge2(2)=sqrt(sum((x1(2,1)-a([6:10],1)).^2));
judge2(3)=sqrt(sum((x1(2,1)-a([11:15],1)).^2));
[E2 q2]=sort(judge2);
class=q2(1)
err=E2(q2(1))

%----- angle_divide.m-----%
clear all
clc
load sample.mat
t1=ChannellR;
for i=1:16201
    i
    sum1=sum(sum(abs(angle(t1([i:i+1499],:))))));
    if sum1<=2.3607e+05&sum1>2.3549e+05
        class(i)=1;
    else if sum1>=2.3494e+05&sum1<=2.3549e+05
        class(i)=2;
    else if sum1>=2.3635e+05&sum1<=2.3720e+05
        class(i)=3;
    else if sum1<2.3494e+05
        class(i)=4;
    else if sum1>2.3720e+05
        class(i)=5;
    end
    end
end
end
end
k=1;
for i=1:16200
    if class(i)~=class(i+1)
        divide(k)=i
    end
end

```

```

changjing(k)=class(i)
k=k+1
end
end

%----- julei_divide.m-----%
clear all
clc
%聚类分析求聚类半径方差
load Sample.mat
t1=ChannellR;
r=real(t1);
r_means=sum(sum(r))/1770000;
i=imag(t1);
i_means=sum(sum(i))/1770000;
X=sqrt(r.^2+i.^2);
%X欧式距离计算
dist_oushi=sqrt((r-r_means).^2+(i-i_means).^2);
k=1;

for j=1:17601
    for i=1:100
        dist(i)=sum(dist_oushi([j:j+99],i))/100;
    end
    dist_std0(k)=std(dist);
    k=k+1;
end
plot(dist_std0,'-*')
grid on

[E1 p1]=sort(dist_std0(1:2000));
point1=p1(2000)
[E2 p2]=sort(dist_std0(4001:6000));
point2=p2(2000)+4000
[E3 p3]=sort(dist_std0(10001:12000));
point3=p3(2000)+10000
[E4 p4]=sort(dist_std0(12001:13000));
point4=p4(1000)+12000
[E5 p5]=sort(dist_std0(13001:15000));
point5=p5(2000)+13000
[E6 p6]=sort(dist_std0(15001:16000));
point6=p6(1000)+15000

%-----angle_divide.m-----%

```

```

clear all
clc
load Sample.mat
t1=ChannelIR;
t=1;
for j=1:100:17700
    for i=j:j+99
        for k=1:100
            sum1(k)=sum(abs(angle(t1(i,k))));
        end
    end
    ang(t)=std(sum1);
    t=t+1;
end
plot(ang,'-o')
grid on

load SampleForTest1.mat
t1=ChannelIR;
t=1;
for j=1:100:200
    for i=j:j+99
        for k=1:100
            sum11(k)=sum(abs(angle(t1(i,k))));
        end
    end
    ang1(t)=std(sum11);
    t=t+1;
end

load SampleForTest2.mat
t1=ChannelIR;
t=1;
for j=1:100:200
    for i=j:j+99
        for k=1:100
            sum12(k)=sum(abs(angle(t1(i,k))));
        end
    end
    ang2(t)=std(sum12);
    t=t+1;

```

```

end

%最小二乘法判别最小匹配最方差
for i=1:176
    ange(i)=(ang1(1)-ang(i))^2+(ang1(2)-ang(i+1))^2;
end
[E,q]=sort(ange);
ange=E(1)
i1=q(1)

for i=1:176
    ange(i)=(ang2(1)-ang(i))^2+(ang2(2)-ang(i+1))^2;
end
[E,q]=sort(ange);
ange=E(1)
i2=q(1)
%-----dist_std_divide.m-----%
clear all
clc
%聚类分析求聚类半径方差
load Sample.mat
t1=ChannellR;
r=real(t1);
r_means=sum(sum(r))/1770000;
i=imag(t1);
i_means=sum(sum(i))/1770000;
X=sqrt(r.^2+i.^2);
%X欧氏距离计算
dist_oushi=sqrt((r-r_means).^2+(i-i_means).^2);
k=1;
for j=1:100:17700
    for i=1:100
        dist(i)=sum(dist_oushi([j:j+99],i))/100;
    end
    dist_std0(k)=std(dist);
    k=k+1;
end
plot(dist_std0,'*r')
grid on

load SampleForTest1.mat
t1=ChannellR;
r=real(t1);
r_means=sum(sum(r))/20000;

```

```

i=imag(t1);
i_means=sum(sum(i))/20000;
X=sqrt(r.^2+i.^2);
%欧式距离计算
dist_oushi=sqrt((r-r_means).^2+(i-i_means).^2);
k=1;
for j=1:100:200
    for i=1:100
        dist(i)=sum(dist_oushi([j:j+99],i))/100;
    end
    dist_std1(k)=std(dist);
    k=k+1;
end

load SampleForTest2.mat
t1=ChannellIR;
r=real(t1);
r_means=sum(sum(r))/20000;
i=imag(t1);
i_means=sum(sum(i))/20000;
X=sqrt(r.^2+i.^2);
%欧式距离计算
dist_oushi=sqrt((r-r_means).^2+(i-i_means).^2);
k=1;
for j=1:100:200
    for i=1:100
        dist(i)=sum(dist_oushi([j:j+99],i))/100;
    end
    dist_std2(k)=std(dist);
    k=k+1;
end

%最小二乘法匹配方差
for i=1:176
    std_diff(i)=(dist_std1(1)-dist_std0(i))^2+(dist_std1(2)-dist_std0(i+1))^2;
end
[E,q]=sort(std_diff);
min_diff_std1=E(1)
i1=q(1)

for i=1:176
    std_diff(i)=(dist_std2(1)-dist_std0(i))^2+(dist_std2(2)-dist_std0(i+1))^2;
end
[E,q]=sort(std_diff);

```

```

min_diff_std2=E(1)
i2=q(1)

%在窗口滑动情形下的最小误差匹配
load Sample.mat
t1=ChannellR;
R=real(t1);
r_means=sum(sum(R))/1770000;
I=imag(t1);
i_means=sum(sum(I))/1770000;
dist_oushi_new=sqrt((R-r_means).^2+(I-i_means).^2);

k=1;
for j=1751:1850          %窗口为 100, 以单位为 1 前后滑动 50 单位
    for i=1:100
        dist_change1(i)=sum(dist_oushi_new([jj+99],i))/100;
    end
    dist_std_new1(k)=std(dist_change1);
    k=k+1;
end

k=1;
for j=1851:1950          %窗口为 100, 以单位为 1 前后滑动 50 单位
    for i=1:100
        dist_change2(i)=sum(dist_oushi_new([jj+99],i))/100;
    end
    dist_std_new2(k)=std(dist_change2);
    k=k+1;
end

for i=1:100
    std_diff_new1(i)=(dist_std1(1)-dist_std_new1(i))^2+(dist_std1(2)-dist_std_new2(i))^2;
end
[E_new1,q_new1]=sort(std_diff_new1);
min_diff_std_new1=E_new1(1)
i_new1=q_new1(1)

%在窗口滑动情形下的最小误差匹配
load Sample.mat
t1=ChannellR;
R=real(t1);
r_means=sum(sum(R))/1770000;

```

```

I=imag(t1);
i_means=sum(sum(I))/1770000;
dist_oushi_new=sqrt((R-r_means).^2+(I-i_means).^2);

k=1;
for j=8851:8950      % 窗口为 100, 以单位为 1 前后滑动 50 单位
    for i=1:100
        dist_change1(i)=sum(dist_oushi_new([jj+99],i))/100;
    end
    dist_std_new1(k)=std(dist_change1);
    k=k+1;
end

k=1;
for j=8951:9050      % 窗口为 100, 以单位为 1 前后滑动 50 单位
    for i=1:100
        dist_change2(i)=sum(dist_oushi_new([jj+99],i))/100;
    end
    dist_std_new2(k)=std(dist_change2);
    k=k+1;
end

for i=1:100
    std_diff_new2(i)=(dist_std2(1)-dist_std_new1(i))^2+(dist_std2(2)-dist_std_new2(i))^2;
end

[E_new2,q_new2]=sort(std_diff_new2);
min_diff_std_new2=E_new2(1)
i_new2=q_new2(1)

%-----picture_divide.m-----%
clear all
clc
load SampleForTest1.mat
t1=ChannellR;
t1_mo=abs(t1);
mo_max=max(max(t1_mo));
t1_mo=t1_mo/mo_max;
k=1:100;
n=1:200;
[k,n]=meshgrid(k,n);
z=k.*n*0+t1_mo([1:200],:);
figure (1)
surf(k,n,z);
% view(90,0)
shading flat

```

```

load SampleForTest2.mat
t1=ChannelIR;
t1_mo=abs(t1);
mo_max=max(max(t1_mo));
t1_mo=t1_mo/mo_max;
k=1:100;
n=1:200;
[k,n]=meshgrid(k,n);
z=k.*n*0+t1_mo([1:200],:);
figure (2)
surf(k,n,z);
% view(90,0)
shading flat
%-----julei_smoothwindow.m-----
clear all
clc
%聚类分析求聚类半径方差
load Sample.mat
t1=ChannelIR;
r=real(t1);
r_means=sum(sum(r))/1770000;
i=imag(t1);
i_means=sum(sum(i))/1770000;
X=sqrt(r.^2+i.^2);
%欧式距离计算
dist_oushi=sqrt((r-r_means).^2+(i-i_means).^2);
k=1;

for j=1:17601
    for i=1:100
        dist(i)=sum(dist_oushi([j:j+99],i))/100;
    end
    dist_std0(k)=std(dist);
    k=k+1;
end
plot(dist_std0,'-o')
grid on

load SampleForTest1.mat
t1=ChannelIR;
r=real(t1);
r_means=sum(sum(r))/20000;
i=imag(t1);

```

```

i_means=sum(sum(i))/20000;
X=sqrt(r.^2+i.^2);
%欧式距离计算
dist_oushi=sqrt((r-r_means).^2+(i-i_means).^2);
k=1;
for j=1:101
    for i=1:100
        dist(i)=sum(dist_oushi([j:j+99],i))/100;
    end
    dist_std1(k)=std(dist);
    k=k+1;
end

load SampleForTest2.mat
t1=ChannellR;
r=real(t1);
r_means=sum(sum(r))/20000;
i=imag(t1);
i_means=sum(sum(i))/20000;
X=sqrt(r.^2+i.^2);
%欧式距离计算
dist_oushi=sqrt((r-r_means).^2+(i-i_means).^2);
k=1;
for j=1:101
    for i=1:100
        dist(i)=sum(dist_oushi([j:j+99],i))/100;
    end
    dist_std2(k)=std(dist);
    k=k+1;
end

%最小二乘法判别最小匹配最方差
for i=1:17600
    std_diff(i)=(dist_std1(1)-dist_std0(i))^2+(dist_std1(2)-dist_std0(i+1))^2;
end
[E,q]=sort(std_diff);
min_diff_std1=E(1)
i1=q(1)

for i=1:17600
    std_diff(i)=(dist_std2(1)-dist_std0(i))^2+(dist_std2(2)-dist_std0(i+1))^2;
end
[E,q]=sort(std_diff);
min_diff_std2=E(1)

```

```

i2=q(1)
%-----angle_smoothwindow.m-----%
clear all
clc
load Sample.mat
t1=ChannelIR;
t=1;
for j=1:17601
    for i=j+99
        for k=1:100
            sum1(k)=sum(abs(angle(t1(i,k))));
        end

    end
    ang(t)=std(sum1);
    t=t+1;
end
plot(ang,'-o')
grid on

load SampleForTest1.mat
t1=ChannelIR;
t=1;
for j=1:101
    for i=j+99
        for k=1:100
            sum11(k)=sum(abs(angle(t1(i,k))));
        end

    end
    ang1(t)=std(sum11);
    t=t+1;
end

load SampleForTest2.mat
t1=ChannelIR;
t=1;
for j=1:101
    for i=j+99
        for k=1:100
            sum12(k)=sum(abs(angle(t1(i,k))));
        end

    end

```

```

ang2(t)=std(sum12);
t=t+1;
end

% 最小二乘法判别最小匹配最方差
for i=1:17600
    ange(i)=(ang1(1)-ang(i))^2+(ang1(2)-ang(i+1))^2;
end
[E,q]=sort(ange);
ange=E(1)
i1=q(1)

for i=1:17600
    ange(i)=(ang2(1)-ang(i))^2+(ang2(2)-ang(i+1))^2;
end
[E,q]=sort(ange);
ange=E(1)
i2=q(1)

```