

# 统计推断在数模转换系统中的应用

——寻求定标工序的优选方案

组号：20 冯柱天 5140309370，陈科泓 5140309115

**摘要：** 本文是上海交通大学电子信息与电气工程学院课程《统计推断在数模、模数转换系统中的应用》的论文。本文主要阐述了统计推断在数模转换系统中的应用，以为某种电子产品模块的传感器的寻求定标工序为研究实例，经过插值或拟合、启发式搜索等一系列过程，借助 MATLAB R2015b 建立传感器的关系曲线模型，最终得到了在数据点较多情况下如何选择适当数量、适当位置的数据点对输入输出特性校准，并保证校准结果的准确性和校准方案的易操作性的定标工序，实现以少量数据反映系统整体特性的效果。

**关键词：** 定标工序，多项式拟合，插值，遗传算法

## Application of Statistical Inference in DA Inverting System

——Seeking the Best Calibration Procedure

**ABSTRACT:** This article is for Course Design-Application of Statistical Inference in AD&DA Inverting System, School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University. In this article we have elaborated the application in application of statistical inference in AD&DA inverting system. Based on seeking for the calibration procedure in a sensor of certain electronic product, we tried polynomial fitting, interpolation and heuristic search which were run on MATLAB R2015b to build the input-output curves of sensors. Finally we get the procedure achieved that using a small amount of data to reflect the features of the overall system on choosing proper numbers and positions of data points to calibrate, at the same time, we can make sure the accuracy of the result and the handle ability of the method.

**Key words:** calibration procedure, polynomial fitting, interpolation, Genetic Algorithm (GA)

## 1 引言

假定有某型投入批量试生产的电子产品，其内部有一个模块，功能是监测某项与外部环境有关的物理量（可能是温度、压力、光强等），但该监测模块中传感器部件的输入输出特性呈明显的非线性。因此在该产品的生产中，我们需要对该产品模块的输入与输出特性进行定标。在定标中，选取的点越多，就越能符合该模块的实际情况。但问题在于在实际生产中，若选取过多的点定标，所花费的成本过高。因此本课题要求为该模块的批量生产设计一种成本合理的传感特性校准（定标工序）方案<sup>[1]</sup>。

## 2 数值计算方法

为了对本课题展开有效讨论，需建立一个数学模型，对问题的某些方面进行必要的描述和限定，之后才可采用数值方法进行计算。

## 2.1 传感器部件特性

我们随机选取了四个传感部件，其输入输出特性大概如图 1-1 所示。可以看出，其特性曲线呈现明显的非线性。对该曲线我们采用曲线拟合的方法，将在 2.2 节中论述。

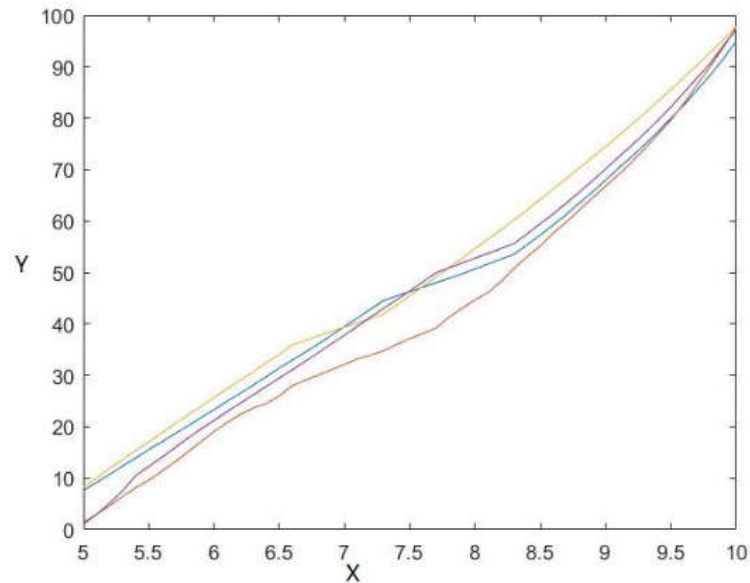


图 1-1 随机选取的四个传感器输出输入特性曲线

## 2.2 拟合方法的选择

### 2.2.1 多项式拟合

科学或工程上可以通过实验等方法获得关于某个问题的若干离散数据，然后依靠数学方法，使用连续函数（也就是曲线）或者更加密集的离散方程尽量逼近（即最小二乘意义上的差别最小化）这些已知离散数据点集，此过程称为拟合。由高等数学知识可知，任意函数均可展开为多项式函数。由上图我们所得到的样本特征，图像的斜率由大到小在到在，有两次明显的凹凸性的变化，其导函数是一个下凹的函数，形状与开口向上的抛物线相似，故我们考虑用三次多项式函数拟合。

在 MATLAB R2015b 中，提供了多种拟合命令，这里我们使用多项式拟合命令 `Polyfit(xdata, ydata, n)`。其中 `xdata`、`ydata` 分别是横、纵轴的数据，`n` 为多项式次数，拟合的数学表达式为 (2-1)：

$$Y = a_n X^n + a_{n-1} X^{n-1} + a_{n-2} X^{n-2} + \dots + a_1 X^1 + a_0 \quad \dots\dots (2-1)$$

### 2.2.2 插值拟合

插值指的在离散数据的基础上补插连续函数，使得这条连续曲线通过全部给定的离散数据点。插值是离散函数逼近的重要方法，利用它可通过函数在有限个点处的取值状况，估算出函数在其他点处的近似值。主要的插值方法有 Lagrange 插值法与 Hermite 插值法，其中后者不仅需要在给定离散点的函数值相同，更要求在这些节点上的导数也相同。

在 MATLAB R2015b 中提供了多种插值方法，其函数为 `vq = interp1(x,y,xq,method)`。其中适合图 1-1 传感器输出输入曲线的插值方法有两种，一种是 'pchip'（分段 Hermite 插值），一种是 'spline'（三次样条插值）。三次样条插值只要求在每个分段区间之内用三次多项式拟合<sup>[2]</sup>，而分段 Hermite 插值还要求拟合函数在节点的导数与原始值相同<sup>[3]</sup>，这样做出的曲线更加光滑。性能上，根据 MATLAB 官方文档的说明，三次样条插值将比分段 Hermite 插值需要更多的计算量和内存空间<sup>[4]</sup>。

我们将分别运用这两种插值方法对课题进行求解，并在第四部分详细比较讨论采用这两种插值方法的不同所带来的定标成本的差异。

### 3 搜索算法

#### 3.1 概论

搜索算法是指利用计算机的高性能运算能力,有目的地检验一个问题之解空间包含的部分或全部可能情况,从而求得问题的优化解的做法。在本课题中,若对取点的方法及其数目做穷举法,其方案数将达到:

$$C_{51}^1 + C_{51}^2 + C_{51}^3 + \cdots + C_{51}^{50} + C_{51}^{51} = 2^{51} - 1 \quad (3-1)$$

显然这个数字已经远远超出了计算机的处理能力,是典型的 NP-hard 问题。因此我们需要采用别的方搜索算法才能将运算量控制在我们目前计算机所能计算的水平内。这种时候,启发式搜索可以被我们派上用场。在搜索问题解空间时,对当前已搜索的位置进行评估,寻找认为最好的下一步搜索方向,从这个(或这些)方向进行搜索直到目标。启发式搜索可以避免简单穷举式搜索效率低(甚至不可完成)的弊端。

在启发式搜索算法中,我们从模拟退火算法和遗传算法中选用了遗传算法来进行取点方案的搜索。

#### 3.2 遗传算法

##### 3.2.1 遗传算法概述

遗传算法(Genetic Algorithm, GA)是计算机科学人工智能领域中用于解决最优化的一种搜索启发式算法,是进化算法的一种。这种启发式通常用来生成有用的解决方案来优化和搜索问题。它最初是借鉴了进化生物学中的一些现象而发展起来的,这些现象包括遗传、突变、自然选择以及杂交等<sup>[5]</sup>。遗传算法的这些性质,已被人们广泛地应用于组合优化、机器学习、信号处理、自适应控制和人工生命等领域。

##### 3.2.1 遗传算法实现

遗传算法的主要过程可分为个体编码、初始化、适应度计算、选择、交叉、变异,并一代代的进行循环,直到达到终止条件(达到最大进化代数)。图 3-1 所示为本问题中采用遗传算法的主要流程图。对此流程图的逐项解释如下:

###### (1) 初始化程序

初始化程序用于初始化程序的基本参数:初代种群个数为 100 个,取点范围为 1 到 50,最大代数为 100 代,最小费用为无穷大。取点个数的范围为 4 到 50 个点。从我们运行得到的可视化结果来看,选择代数为 100 是较为理想的选择,此时成本已收敛较多的代数且我们最终得到的成本值没有较大的波动。

###### (2) 初始化初代种群与种群编码

对种群采用十进制编码,每一个十进制数就代表其为取点方案中的第几个点。初始时采用 MATLAB 的随机函数生成 100 个初代种群。

###### (3) 成本计算

由于选择,交叉,变异所生成的个体可能会出现重复取点的情况,所以在成本计算之前先判断是否有重复取点的情况,若有则将其成本设为一个较大的值,在本程序中我们将其设为 400000,则此个体的适应度将极低,其存活几率极低。

若没有出现重复取点,则按照选定的该种取点方式,分别对这 400 个样品取选定的点的数据,按 2.2 节中所讨论的三种拟合方法进行曲线拟合。得到拟合曲线之后,我们回到每个样品的 50 个数据,由拟合曲线计算出每一个样品中误差成本的和(式 3-2),再与取点的成本进行相加(式 3-3),对每一个样品的误差成本求和,最后平均到 400 个样品(式 3-4),就得到每一种取点方式的平均成本。

其中,据本课程规定,误差成本以及取点成本的计算规则以及数学表达式如下:

###### ● 单点定标误差成本

$$s_{i,j} = \begin{cases} 0 & \text{if } |\hat{y}_{i,j} - y_{i,j}| \leq 0.4 \\ 0.1 & \text{if } 0.4 < |\hat{y}_{i,j} - y_{i,j}| \leq 0.6 \\ 0.7 & \text{if } 0.6 < |\hat{y}_{i,j} - y_{i,j}| \leq 0.8 \\ 0.9 & \text{if } 0.8 < |\hat{y}_{i,j} - y_{i,j}| \leq 1 \\ 1.5 & \text{if } 1 < |\hat{y}_{i,j} - y_{i,j}| \leq 2 \\ 6 & \text{if } 2 < |\hat{y}_{i,j} - y_{i,j}| \leq 3 \\ 12 & \text{if } 3 < |\hat{y}_{i,j} - y_{i,j}| \leq 5 \\ 25 & \text{if } |\hat{y}_{i,j} - y_{i,j}| > 5 \end{cases} \quad (3-2)$$

单点定标误差的成本按式 (3-2) 计算, 其中  $y_{i,j}$  表示第  $i$  个样本之第  $j$  点  $Y$  的实测值,

$\hat{y}_{i,j}$  表示定标后得到的估测值 (读数), 该点的相应误差成本以符号  $s_{i,j}$  记。

- 单点测定成本

实施一次单点测定的成本以符号  $q$  记。本课题指定  $q=12$ 。

- 某一样本个体的定标成本

$$S_i = \sum_{j=1}^{51} s_{i,j} + q \cdot n_i \quad (3-3)$$

对样本  $i$  总的定标成本按式 (3-3) 计算, 式中  $n_i$  表示对该样本个体定标过程中的单点测定次数。

- 校准方案总成本

按式 (3-4) 计算评估校准方案的总成本, 即使用该校准方案对标准样本库中每个样本个体逐一定标, 取所有样本个体的定标成本的统计平均。

$$C = \frac{1}{M} \sum_{i=1}^M S_i \quad (3-4)$$

总成本较低的校准方案, 认定为较优方案。

#### (4) 适应度计算

适应度函数将会计算个体的优劣程度, 从而决定其遗传机会的大小。适应度越高的个体, 在选择的过程中其遗传的机会越大。我们的适应度将成本最小的个体取为 100, 成本最大的个体取为 0, 并且按比例根据成本计算每个个体的适应度。

#### (5) 选择

选择的过程是将适应度高的个体将更有可能在选择中被保留下来。我们采用的是适应度比例方法, 每个个体的生存概率与其适应度成正比, 我们取选择的概率为 0.7, 采用轮盘赌的方法对个体进行选择, 同时我们也将种群中最优的一个个体强制加入下一代个体中, 以确保最优的个体一定能被选择进入下一代。

#### (6) 交叉

模拟自然中的交配现象, 将父母的数据的一部分进行交换, 从而产生两个新个体。这是遗传算法起核心作用的部分。我们选择交叉的概率为 0.7, 选取 0.7 的情况下交叉概率较大,

而且成本收敛的速度在可接受的范围。交叉生成的下一代的个数为  $2 \times 0.7 \times$  上一代个体个数，每一个生成的个体都由两个随机的上一代个体构成。模拟生物中的交叉，我们将这两个上一代个体被一个随机生成的交叉点分为两部分，取一个个体的前一部分和另一个个体的后一部分组成下一代的个体。

### (7) 变异

变异运算是模拟自然中的基因突变，对一些数据进行小概率的改变。我们选取变异概率为 0.3，即一个随机位置上的值变成另一个随机的值的概率为 0.3。

经过以上步骤后，我们就产生了下一代新的种群，如此循环下去，直到预先设定的遗传代数。这样便可以得到最终结果。其中，在遍历取点个数时，如果之前的得到的最小成本已经小于现在的取点成本时（如若目前最小成本为 85，当遍历到取 8 个点的取点方案时，循环将自动跳出，因为若取 8 个点则取点成本就已达 96，比目前的最小成本高）也同样跳出循环。

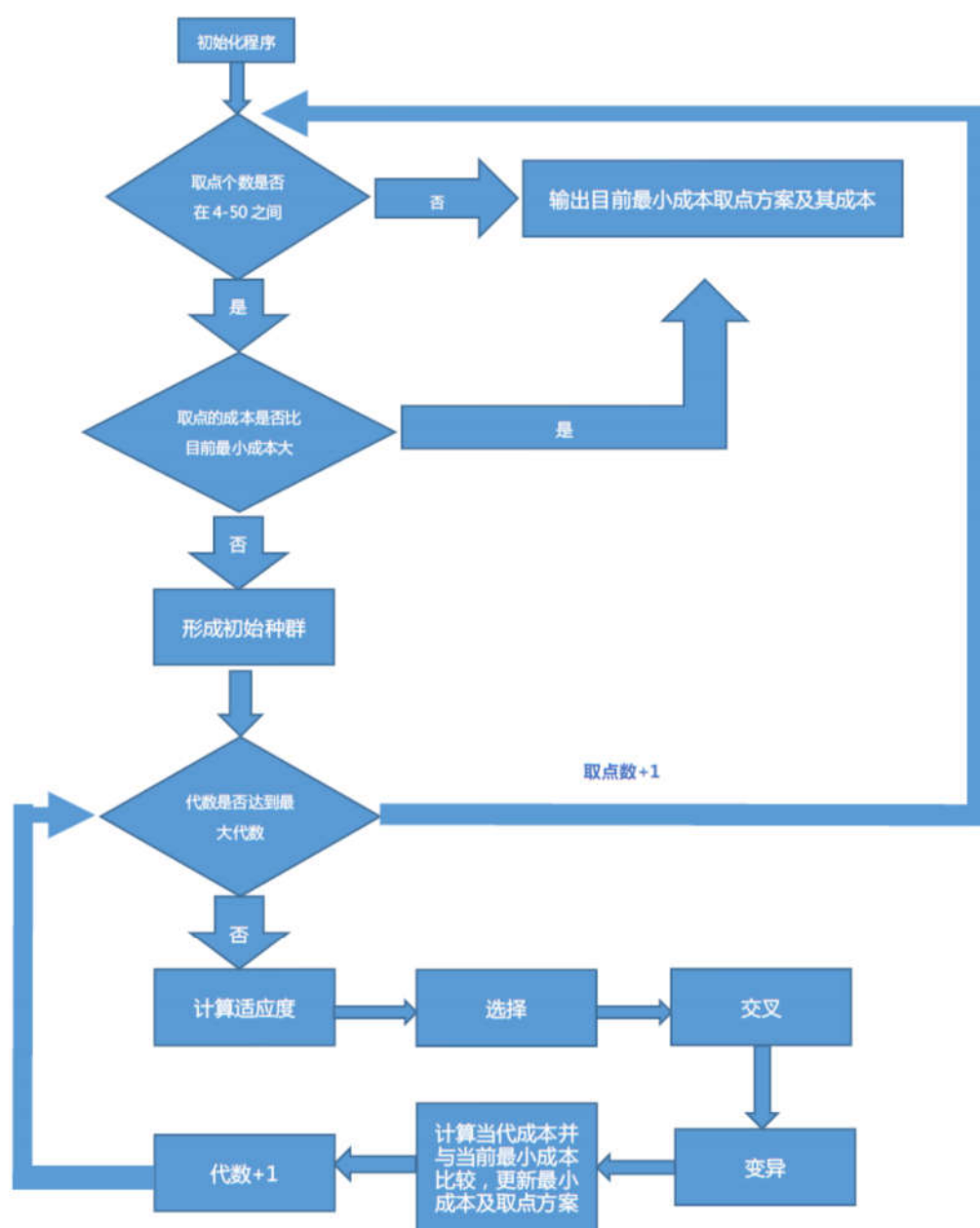


图 3-1 遗传算法的主要流程图

### 3.2.3 遗传算法的优缺点

遗传算法不存在求导和函数连续性的限定；具有内在的隐并行性和更好的全局寻优能力；采用概率化的寻优方法，能自动获取和指导优化的搜索空间，自适应地调整搜索方向，不需要确定的规则。但与此同时，遗传算法有可能收敛于局部最优，而不能达到全局最优。遗传算法通常的效率比其他传统的优化方法（如模拟退火算法）低，且遗传算法容易出现过早收敛。对算法的精度、可行度、计算复杂性等方面，还没有有效的定量分析方法。因此我们在确定程序以后，将采取多次运行的方法，以确保不会因为偶然因素产生较大的偏差。

## 4 传感部件输入输出特性定标方案设计

在搜索算法确定采用遗传算法的情况下，下文主要讨论的是在此基础上研究不同的拟合方法和取点方案的选择对最终平均成本的影响。现将多次试验的结果列成表格记录如下：

表4-1 三次多项式(polyfit)拟合的结果

试验次数	试验结果	校准方案总成本
1	3 12 24 35 48	114.82
2	3 13 24 35 47	115.28
3	4 15 27 38 49	114.20

表4-2 样条插值(spline)拟合的结果

试验次数	试验结果	校准方案总成本
1	2 10 21 30 40 49	95.49
2	3 11 22 31 43 50	94.92
3	3 12 22 31 43 50	94.90
4	3 12 22 31 43 50	94.90
5	3 12 22 31 43 50	94.90
6	3 10 20 27 35 44 50	95.34

表4-3 分段三次Hermite插值(pchip)的结果

试验次数	试验结果	校准方案总成本
1	4 17 27 36 48	85.49
2	4 16 26 35 48	84.75
3	4 16 26 35 48	84.75
4	4 16 26 35 48	84.75
5	4 16 26 35 48	84.75
6	3 14 24 34 47	85.15

### 4.1 拟合方法的选择

从上述表格我们可以看出，三次多项式在取点数目较小的情况下，仍然具有较大的误差，而且成本较高。因此，我们通过对比与两种插值方法所得结果对比，可知三次多项式拟合不是本课题最好的拟合方法。从时间角度上来看，polyfit 函数运行的时间是 interp1 函数的十倍左右，我们插值的结果大概只需要 15-20min，而多项式拟合需要 1.5hrs 左右的运行时间。因此我们抛弃了三次多项式的拟合方法。

从两种插值的方法来看，分段 Hermite 插值的结果比样条插值的结果成本更低，而且根据 MATLAB 帮助文档的说明，样条插值需要比分段 Hermite 插值更多的内存和计算量。因此，我们认为分段三次 Hermite 插值是本次课题最为理想的插值拟合方法

### 4.2 取点方案的选择

我们的源程序中，取点数目为 4 到 50。从两种插值方法得到的结果来看，5-7 点应为较

为理想的取点数目，此时达到了总成本中误差与取点成本的平衡。无论是哪一种插值拟合方法，从多次程序的运行结果来看，取 8 点的方案由于取点成本已经大于目前成本的最小值，因此程序跳出循环，此时运行结束输出结果。

对于样条插值的方法，我们在最后一次运行中得到了一个取 7 点的方案，而且误差成本在可以接受的范围之内。这让我们觉得，随着技术的进步，单点测定的成本逐渐下降，采用 7 点的取点方案将能获得更准确的特性曲线，因此该方案具有更普遍的意义。

对于分段 Hermite 插值，若我们选定取点的个数，从多次运行的结果来看，实际上取 5 点或取 6 点的区别并不大，误差仅在 84-86 之间波动（见表 4-4）。而且在测定成本增加了 12（取点数目增加 1 点）的情况下，误差成本减少了近 10.5，因此我们认为，同样随着技术的进步，单点测定成本的下降，取较多点的方案更大程度上能减少误差。

表4-4 分段三次Hermite插值(pchip)不同取点数的结果

取点数	试验结果	校准方案总成本
6	4 15 24 31 39 49	86.61
5	4 16 26 35 48	84.75

4.3 定标方案的确定

综合上述研究，我们可以初步认为，在单一取点方案的情况下，取 5 点（4、16、26、35、48）并采取分段 Hermite 插值的定标工序，平均定标成本大约在 84.75 左右，在可接受的范围之内。选定分段 Hermite 插值，运行程序并将每一代的成本作图，可以得到下图：

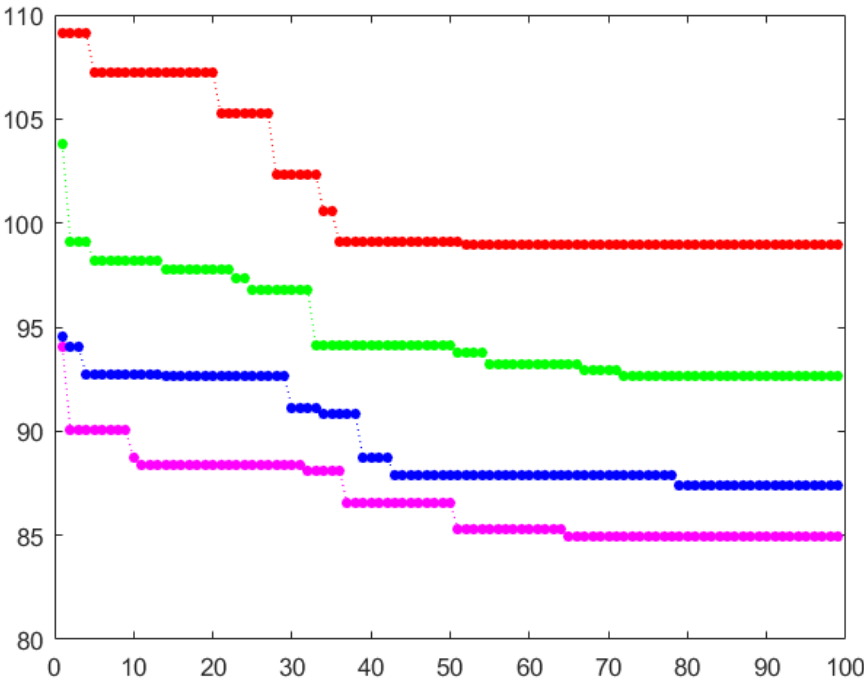


图 4-1 分段三次 Hermite 插值(pchip)下不同取点数的成本变化曲线

图 4-1 中不同的颜色代表不同的取点数目，从高到低分别为取点数为 4、7、6、5 的情况。可以看到，随着遗传代数的增加，成本变化的幅度逐渐减小，在 70 代以后成本趋于收敛于某个固定值，波动较小。因此我们遗传算法中的各项参数是较为合适的。

取 6 点的情况下，尽管总成本会上升到 86.61 左右，但我们认为，随着技术的进步，单点测定成本的下降，取 6 点的方案能获得更准确的曲线，因此其更具有普遍意义。

## 5 取点方案的进一步改进

在使用遗传算法得到上述 4.3 节所述的定标方案后我们发现，多次运行采用分段三次 Hermite 插值(pchip)的程序，其最终结果一般集中在表 4-4 中的两个结果，即取六个点 [4,15,24,31,39,49]的成本为 86.61 的方案，取五个点[4,16,26,35,48]为 84.75 的方案。虽然多取一个点取点的成本会高 12，但实际上最后的成本 6 个点只比 5 个点多了 1.86。所以 6 个点带来的误差成本的减小是非常明显的。

所以我们考虑这种改进方法，是否可以对于某些情况采用 5 个点拟合，对于另外的情况采用 6 个点拟合。在多次运行程序之后，我们发现这两种取点方式都包含第 4 个点，因此我们推测是否可以根据第 4 个点的值来判断采用 5 个点还是 6 个点拟合。于是我们统计了这 400 个传感器的数据，计算它们用 5 个点拟合的误差成本和 6 个点拟合的误差成本。若某一点的 6 个点的误差成本加上 12 仍比 5 个点的误差成本小，则这一点采用 6 个点更好，反之，则使用 5 个点拟合。

我们对这 400 组数据进行处理，在第四个点的位置，其 Y 值最大为 17.35，最小值为 3.79，其差值为 13.56。据此我们对这 400 组数据在第四个点的值进行分段，分为 140 段，对在每段的数据进行统计，设在一段内有 n 组数据（n 个传感器），第 i 组的 6 个点的误差成本为  $A_i$ ，5 个点的误差成本  $B_i$ 。若式 (5-1)

$$\sum_{i=0}^n (A_i + 12 - B_i) < 0 \quad (5-1)$$

则取 6 个点，反之，取 5 个点。这样做的意义在于：第 4 个点的值属于某一段时，在统计学的概率上，这一组数据更趋向于采用某一种取点方式（取 5 个点或取 6 个点）。即当第 4 点的值落在的时候，我们就可以决定是采用哪种取点数。

建立在这种思路，我们写了 FiveOrSix 的函数，只需向其传递第 4 个点的值，其会在已有的分段的数据中进行查找，并返回应该使用的取点方案。若其返回 5，则取 5 个点，若返回 6，则取 6 个点。因为第 4 个点是这两种取点方式都需要的，所以不需要额外的成本。

采用这样的取点方案后，成本计算会对不同的数据采用不同的取点方案，所以我们写了对应的成本计算函数 FuzzyCosts。其对于一组数据，先取第 4 个点的值调用函数 FiveOrSix 计算采用哪种取点方案，再根据取点方案进行拟合计算其误差成本，且取点成本为 60（取 5 点）或 72（取 6 点），对每一个传感器的两项成本相加得每一个传感器的和，最后对 400 组的和求平均，得到其平均成本。

综合上述，我们运行 Fuzzycost 对 400 个传感器采用不同的取点方法（即取 6 个点时为 [4,15,24,31,39,49]，取五个点时为[4,16,26,35,48]），采用此种取点方案我们最终可以得到 83.0505 的取点成本，相比单一的取点方法可以减少 1.7 左右的成本，而且对取 6 点的传感器，得到的曲线更为精确。只需根据第 4 点的值判断对该传感器是取 6 个点还是 5 个点，并且可使总体方案下降，因此这种改进具有实际的意义。

## 5 附录

### 5.1 程序代码

#### 5.1.1 程序主入口代码 main.m

```
infile = csvread('20150915dataform.csv');  
  
num = 100;  
datamax = 50;  
maxgen = 100;
```



```

main_min = inf;
for num_of_points = 4:50
    if num_of_points * 12 > main_min
        break;
    end
    parents = init(num,datamax,num_of_points);
    money = costs(infile,parents);

    gen = 1;
    gen_min = inf;
    while(gen<maxgen)
        fitness = ranking(money);
        parents = select(parents,fitness,0.7);
        parents = recomb(parents,0.7);
        parents = mut(parents,datamax,0.3);
        money = costs(infile,parents);
        [Y,I] = min(money);
        fprintf('Num of points: %d\n',num_of_points);
        fprintf('Generation: %d\n',gen);
        fprintf('Min cost: %5.2f\n',Y);
        if Y<=gen_min
            gen_min = Y;
            gen_min_data = parents(I,:);
        end
        gen = gen+1;
    end
    fprintf('The final scheme is(%d
points):\n',num_of_points);
    fprintf('Min cost: %5.2f\n',gen_min);
    disp(gen_min_data);
    if gen_min<=main_min
        main_min_data = gen_min_data;
        main_min = gen_min;
    end
end
fprintf('The final scheme is:\n');
disp(main_min_data);
fprintf('The final cost is :%5.2f\n',main_min);

```

### 5.1.2 计算成本代码 costs.m

```

function money = costs(infile,data)
    size_of_data = length(data(:,1));
    num_of_points = length(data(1,:));
    money = zeros(1,size_of_data);

```

```

xx = [5:0.1:10];
xdata = zeros(1,num_of_points);
ydata = zeros(1,num_of_points);
for k = 1:size_of_data
    for i=1:400
        yy = infile(i*2,:);
        b = data(k,:);
        flag = 1;
        for i1 = 1:num_of_points-1
            for i2 = i1+1:num_of_points
                if b(i1)==b(i2)
                    flag = 0;
                end
            end
        end
        if flag==0
            money(1,k) = 400000;
            break;
        end
        for j = 1:num_of_points
            xdata(1,j) = xx(data(k,j));
            ydata(1,j) = yy(1,data(k,j));
        end
        estiydata =
interp1(xdata,ydata,xx,'pchip');
        sum = 0;
        for j = 1:51
            if abs(estiydata(j)-yy(j))<=0.4
                continue;
            end
            if abs(estiydata(j)-yy(j))<=0.6
                sum = sum + 0.1;
                continue;
            end
            if abs(estiydata(j)-yy(j))<=0.8
                sum = sum + 0.7;
                continue;
            end
            if abs(estiydata(j)-yy(j))<=1
                sum = sum + 0.9;
                continue;
            end
            if abs(estiydata(j)-yy(j))<=2
                sum = sum + 1.5;
            end
        end
    end
end

```

```

        continue;
    end
    if abs(estiydata(j)-yy(j))<=3
        sum = sum + 6;
        continue;
    end
    if abs(estiydata(j)-yy(j))<=5
        sum = sum + 12;
        continue;
    end
    if abs(estiydata(j)-yy(j))>5
        sum = sum + 25;
        continue;
    end
end
end
    money(1,k) = money(1,k)+sum +
12*num_of_points;
    end
    money(1,k) = money(1,k)/400;
end
    money = money';
end

```

#### 5.1.3 初始化程序代码 init.m

```

function parents = init(num,datamax,num_of_points)
    parents = zeros(num,num_of_points);
    for i = 1:num
        temp = randperm(datamax);
        parents(i,:) = temp(1:num_of_points);
    end
end

```

#### 5.1.4 计算适应度代码 ranking.m

```

function fitness = ranking(costs)
    fmin = min(costs);
    fmax = max(costs);
    if fmin==fmax
        fitness = 100*costs/fmin;
    else
        num_of_data = length(costs);
        fitness = zeros(num_of_data,1);
        for i = 1:num_of_data
            fitness(i) = 100*(fmax-costs(i))/(fmax-
fmin);
        end
    end
end

```

```

    end
end
function children = select(parents,fitness,GGAP)
    num_of_points = length(parents(1,:));
    num_of_parents = length(fitness);
    num_of_children = fix(num_of_parents * (1-GGAP));
    if num_of_children<15
        num_of_children = 20;
    end
    children = zeros(num_of_children,num_of_points);
    p = zeros(num_of_parents,1);
    s = sum(fitness);
    p(1) = fitness(1)/s;
    for i = 2:num_of_parents
        p(i) = p(i-1)+fitness(i)/s;
    end
    for i = 1:num_of_children
        goal = rand();
        for j =1:num_of_parents
            if p(j)>=goal
                children(i,:) = parents(j,:);
                break;
            end
        end
    end
    end
    [~,I] = max(fitness);
    children = [children',parents(I,:)']';
end
function children = select(parents,fitness,GGAP)
    num_of_points = length(parents(1,:));
    num_of_parents = length(fitness);
    num_of_children = fix(num_of_parents * (1-GGAP));
    if num_of_children<15
        num_of_children = 20;
    end
    children = zeros(num_of_children,num_of_points);
    p = zeros(num_of_parents,1);
    s = sum(fitness);
    p(1) = fitness(1)/s;
    for i = 2:num_of_parents
        p(i) = p(i-1)+fitness(i)/s;
    end
    for i = 1:num_of_children
        goal = rand();

```

```

        for j =1:num_of_parents
            if p(j)>=goal
                children(i,:) = parents(j,:);
                break;
            end
        end
    end
    [~,I] = max(fitness);
    children = [children',parents(I,:)']';
end

```

#### 5.1.5 选择代码 select.m

```

function children = select(parents,fitness,GGAP)
    num_of_points = length(parents(1,:));
    num_of_parents = length(fitness);
    num_of_children = fix(num_of_parents * (1-GGAP));
    if num_of_children<15
        num_of_children = 20;
    end
    children = zeros(num_of_children,num_of_points);
    p = zeros(num_of_parents,1);
    s = sum(fitness);
    p(1) = fitness(1)/s;
    for i = 2:num_of_parents
        p(i) = p(i-1)+fitness(i)/s;
    end
    for i = 1:num_of_children
        goal = rand();
        for j =1:num_of_parents
            if p(j)>=goal
                children(i,:) = parents(j,:);
                break;
            end
        end
    end
    [~,I] = max(fitness);
    children = [children',parents(I,:)']';
end

```

#### 5.1.6 交叉代码 recomb.m

```

function children = recomb(parents,GGAP)
    children = parents;
    num_of_parents = length(parents(:,1));
    num_of_points = length(parents(1,:));

```

```

num = fix(2*GGAP*num_of_parents);
for num_of_children = 1:num
    i = unidrnd(num_of_parents);
    j = unidrnd(num_of_parents);
    if i~=j
        n = unidrnd(num_of_points);
        temp = parents(i,:);
        temp(n:num_of_points) =
parents(j,n:num_of_points);
        children = [children',temp']';
    end
end
end
end

```

#### 5.1.7 变异代码 mut.m

```

function children = mut(parents,datamax,GGAP)
    children = parents;
    num_of_parents = length(parents(:,1));
    num_of_points = length(parents(1,:));

    for i = 1:num_of_parents
        goal = rand();
        if goal<GGAP
            n = unidrnd(num_of_points);
            data = unidrnd(datamax);
            temp = parents(i,:);
            temp(n) = data;
            children = [children',temp']';
        end
    end
end
end

```

#### 5.1.8 选取 5 点或 6 点计算成本 Fuzzycosts.m

```

function money = FuzzyCosts()
    infile = csvread('20150915dataform.csv');
    money = 0;
    xx = [5:0.1:10];

    for i=1:400
        yy = infile(i*2,:);
        num_of_points = FiveOrSix(yy(4));
        if(num_of_points==5)
            data = [4,16,26,35,48];
        else

```

```

        data = [4,15,24,31,39,49];
    end
    xdata = zeros(1,num_of_points);
    ydata = zeros(1,num_of_points);
    for j = 1:num_of_points
        xdata(1,j) = xx(data(j));
        ydata(1,j) = yy(1,data(j));
    end
    estiydata = interp1(xdata,ydata,xx,'pchip');
    sum = 0;
    for j = 1:51
        if abs(estiydata(j)-yy(j))<=0.4
            continue;
        end
        if abs(estiydata(j)-yy(j))<=0.6
            sum = sum + 0.1;
            continue;
        end
        if abs(estiydata(j)-yy(j))<=0.8
            sum = sum + 0.7;
            continue;
        end
        if abs(estiydata(j)-yy(j))<=1
            sum = sum + 0.9;
            continue;
        end
        if abs(estiydata(j)-yy(j))<=2
            sum = sum + 1.5;
            continue;
        end
        if abs(estiydata(j)-yy(j))<=3
            sum = sum + 6;
            continue;
        end
        if abs(estiydata(j)-yy(j))<=5
            sum = sum + 12;
            continue;
        end
        if abs(estiydata(j)-yy(j))>5
            sum = sum + 25;
            continue;
        end
    end
    money = money+sum + 12*num_of_points;

```

```

    end
    money = money/400;
end

```

#### 5.1.9 选取 4 点或 5 点 FiveOrSixs.m

```

function ForS = FiveOrSix(num4)
    re = [-10.5,0,0,0,0,0,0,8.8,0,23.7,-3.5,0,-14.6,-
8.8,-2.9,-22.5,0,14.1,-13.2,-20,-26,-4.3,19.8,-
9.1,11.1,10.9,-5.2,6.1,4.1,-8.9,-12.9,-15.9,-36.3,-
15.9,0.9,-39,-27.6,-17.9,56.3,72.7,0,-5.9,-37.4,33,-
19.1,44.3,-53.5,37.7,36.8,-39.1,7.2,-37.8,-6.9,-18.5,-
4.1,-28.7,30.9,-37.8,-6.4,-9.3,-21,-9.2,-72.2,-
43.6,11.7,-6,-13.9,-38.7,-49.5,-7.8,5.8,-9.3,-13,-
13.9,-21.2,14.7,-0.6,17.9,25.2,-32.5,16.1,7.1,-17.8,-
20.6,-13.4,1.7,19.6,0,-55.8,-14.1,-10,-14.5,-
50.2,8.4,11.9,-20.5,-9.2,4.2,-34.4,-5.4,-12.2,-29.9,-
15.9,-8.3,-3.4,-11,2.8,13.9,-9.5,4.5,32.6,-10.4,0,-
15.4,0,27.9,-
3.4,18.3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-
16.9,0,0,0,0]';
    n = fix((num4-3.79)*10)+1;
    if(re(n)>0)
        ForS = 6;
    else
        ForS = 5;
    end
end

```

#### 5.1.10 成本计算标准函数（文中取 6 点的情况）test\_ur\_answer.m

%%%%%% 答案检验程序 2015-11-04 %%%%%%%

```

my_answer=[4,15,24,31,39,49] ;%把你的选点组合填写在此

my_answer_n=size(my_answer,2);

% 标准样本原始数据读入

minput=dlmread('20150915dataform.csv');
[M,N]=size(minput);
nsample=M/2; npoint=N;
x=zeros(nsample,npoint);
y0=zeros(nsample,npoint);
y1=zeros(nsample,npoint);
for i=1:nsample
    x(i,:)=minput(2*i-1,:);

```



```

        y0(i,:)=minput(2*i,:);
end
my_answer_gene=zeros(1,npoint);
my_answer_gene(my_answer)=1;

% 定标计算

index_temp=logical(my_answer_gene);
x_optimal=x(:,index_temp);
y0_optimal=y0(:,index_temp);
for j=1:nsample

    % 请把你的定标计算方法写入函数mycurvefitting

y1(j,:)=mycurvefitting(x_optimal(j,:),y0_optimal(j,:))
;
end

% 成本计算

Q=12;
errabs=abs(y0-y1);

le0_4=(errabs<=0.4);
le0_6=(errabs<=0.6);
le0_8=(errabs<=0.8);
le1_0=(errabs<=1);
le2_0=(errabs<=2);
le3_0=(errabs<=3);
le5_0=(errabs<=5);
g5_0=(errabs>5);

sij=0.1*(le0_6-le0_4)+0.7*(le0_8-le0_6)+0.9*(le1_0-
le0_8)+1.5*(le2_0-le1_0)+6*(le3_0-le2_0)+12*(le5_0-
le3_0)+25*g5_0;
si=sum(sij,2)+Q*ones(nsample,1)*my_answer_n;
cost=sum(si)/nsample;

% 显示结果

fprintf('\n经计算，你的答案对应的总体成本为%5.2f\n',cost);

```

#### 5.1.11 成本计算标准函数（文中取 5 点的情况）test\_ur\_answer2.m

%%%%%%%% 答案检验程序 2015-11-04 %%%%%%%%%

my\_answer=[4,16,26,35,48];%把你的选点组合填写在此

```

my_answer_n=size(my_answer,2);

% 标准样本原始数据读入

minput=dlmread('20150915dataform.csv');
[M,N]=size(minput);
nsample=M/2; npoint=N;
x=zeros(nsample,npoint);
y0=zeros(nsample,npoint);
y1=zeros(nsample,npoint);
for i=1:nsample
    x(i,:)=minput(2*i-1,:);
    y0(i,:)=minput(2*i,:);
end
my_answer_gene=zeros(1,npoint);
my_answer_gene(my_answer)=1;

% 定标计算

index_temp=logical(my_answer_gene);
x_optimal=x(:,index_temp);
y0_optimal=y0(:,index_temp);
for j=1:nsample
    % 请把你的定标计算方法写入函数mycurvefitting

    y1(j,:)=mycurvefitting(x_optimal(j,:),y0_optimal(j,:))
;
end

% 成本计算

Q=12;
errabs=abs(y0-y1);

le0_4=(errabs<=0.4);
le0_6=(errabs<=0.6);
le0_8=(errabs<=0.8);
le1_0=(errabs<=1);
le2_0=(errabs<=2);
le3_0=(errabs<=3);
le5_0=(errabs<=5);
g5_0=(errabs>5);

sij=0.1*(le0_6-le0_4)+0.7*(le0_8-le0_6)+0.9*(le1_0-

```

```

1e0_8)+1.5*(1e2_0-1e1_0)+6*(1e3_0-1e2_0)+12*(1e5_0-
1e3_0)+25*g5_0;
si=sum(sij,2)+Q*ones(nsample,1)*my_answer_n;
cost=sum(si)/nsample;

% 显示结果

fprintf('\n经计算，你的答案对应的总体成本为%5.2f\n',cost);

```

#### 5.1.12 成本计算标准函数中的 mycurvefitting.m

```

function y1 = mycurvefitting( x_premea,y0_premea )

x=[5.0:0.1:10.0];

% 将你的定标计算方法写成指令代码，以下样式仅供参考

y1=interp1(x_premea,y0_premea,x,'pchip');
end

```

## 5.2 词汇表

### 拟合

科学或工程上可以通过实验等方法获得关于某个问题的若干离散数据。依靠数学方法，使用连续函数（也就是曲线）或者更加密集的离散方程尽量逼近（即最小二乘意义上的差别最小化）这些已知离散数据点集，此过程称为拟合。

### 插值（内插）

通常有两种含义。

一是指曲线必须通过若干已知离散数据点的一种拟合；

二是指利用拟合得到的函数曲线在某区间内的值，作为原本数值未知的点的近似取值。

本课程中，“插值”一词主要指上述第二种含义。

### 搜索算法

利用计算机的高性能运算能力，有目的地检验一个问题之解空间包含的部分或全部可能情况，从而求得问题的优化解的做法。

### 启发式搜索

在搜索问题解空间时，对当前已搜索的位置进行评估，寻找认为最好的下一步搜索方向，从这个（或这些）方向进行搜索直到目标。启发式搜索可以避免简单穷举式搜索效率低（甚至不可完成）的弊端。

### 课程自定义词汇：单点测定，单点测定成本

在本课题中，对 X 取某一定值时 Y 的真实取值进行测量确定的过程。完成单点测定要付出一定成本，称为单点测定成本。

### 课程自定义词汇：单点定标误差成本

在本课程中，系统完成定标后，对物理量 Y 的单点读数（即系统对 Y 的估测值）与该点 Y 实测值之间存在误差，按一定规则把该误差折算为一种成本。

### 5.3 符号

$X$	传感部件的输出电压
$Y$	传感部件的输入，即被监测物理量
$\hat{Y}$	监测模块的输出，即监测模块将传感部件输出 $X$ 所转换成的读数（对 $Y$ 的估测值）
$y_{i,j}$	第 $i$ 个样本之第 $j$ 点对应的 $Y$ 实测值（来自标准样本数据库）
$\hat{y}_{i,j}$	第 $i$ 个样本之第 $j$ 点对应的 $Y$ 估测值
$s_{i,j}$	第 $i$ 个样本之第 $j$ 点对应的单点定标误差成本
$q$	单点测定成本
$n_i$	对样本 $i$ 定标过程中的单点测定次数
$S_i$	对样本 $i$ 的定标成本
$M$	标准样本数据库中的样本总数
$C$	基于标准样本数据库评价一个校准方案，算得的该方案总体成本

## 6 参考文献

- 【1】 上海交通大学电子工程系，“统计推断”课程设计的要求 V2.2 2015-9-22.[EB/OL]. [2015-09-22].<ftp://202.120.39.48>.
- 【2】 样条插值-维基百科[G/OL].  
<https://zh.wikipedia.org/wiki/%E6%A0%B7%E6%9D%A1%E6%8F%92%E5%80%BC>
- 【3】 埃尔米特插值-维基百科[G/OL].  
<https://zh.wikipedia.org/wiki/%E5%9F%83%E5%B0%94%E7%B1%B3%E7%89%B9%E6%8F%92%E5%80%BC>
- 【4】 The MathWorks Inc, MATLAB R2015b Help Documents[G/OL].  
<http://cn.mathworks.com/help/matlab/index.html>
- 【5】 遗传算法-维基百科[G/OL].  
<https://zh.wikipedia.org/wiki/%E9%81%97%E4%BC%A0%E7%AE%97%E6%B3%95>
- 【6】 上海交通大学电子工程系，统计推断在数模转换系统中的应用讲义与参考资料 [EB/OL]. [2015-09-24].<ftp://202.120.39.48>.
- 【7】 上海交通大学电子工程系，统计推断-第 nn 组(组长姓名)课程设计报告\_格式模板含参考他人报告申明. [EB/OL]. [2015-09-24].<ftp://202.120.39.48>.

## 7 致谢

在本学期的统计推断课程中，我们小组初步掌握了应用 MATLAB 进行基本的程序编写，并学到了一定的算法设计、数学建模、统计分析技巧，这门课程使我们受益匪浅！

最后感谢袁焱老师和助教老师在面谈和上课时对我们遇到的问题加以指导，使我们成功地完成了本次课题！