

# 统计推断在数模转换系统中的应用

组号：51

姓名：孙欣 学号：5130309250，姓名：李昊轩 学号：5130309252

**摘要：**统计推断学是一类重要的数理统计方法，在各种自然科学和社会科学研究、工程技术等领域都有广泛的应用。在本课程中，我们使用概率论与数理统计中的部分知识以及遗传算法在 MATLAB 中的应用，寻求某产品内部监测模块校准工序的优化方案。所需样本由老师提供，共计 469 组，每组样本观测点 51 个。我们需要确定一种拟合方法根据每组数据尽可能少的观测数据点以达到提高工效的目的来确定系统特性曲线并努力追求与原始数据曲线的精确重合。其中拟合的好坏由所给的成本函数确定。

**关键词：**MATLAB，统计推断，数模转换，三次样条插值，遗传算法

## Statistical inference in the DAC system application

**ABSTRACT:** Statistical inference learning is a kind of important mathematical statistics method, is widely applied in the fields of all kinds of natural science and social science research and engineering technology. In this course, we use the application of probability theory and mathematical statistics knowledge and part of MATLAB genetic algorithm in the optimization scheme for a product, the internal monitoring module calibration procedure. The required samples are provided by the teacher, a total of 469 groups, each group sample observation point 51. We need to determine a fitting method for observation data points according to the data of each group as less as possible to improve the work efficiency of the objective to determine the precise coincidence system characteristic curve and pursue it with the original data curve. The fitting quality in it is determined by the cost function that has been given.

**Key words:** MATLAB, Statistical inference, Digital to analog conversion, Three times spline interpolation, Genetic algorithm

## 1 引言

统计推断学是在概率论的基础上依据样本的有关数据和信息，对未知总体的质量特性参数做出合理的判断和估计。统计推断有着广泛的应用，几乎遍及所有的科学领域，在本课程中，我们应用这种方法研究模型产品内部的监测模块，寻求校准工序的优化方案。

## 2 数据统计与初步分析

### 2.1 问题引入

我们要为某型产品内部的一个监测模块，寻求校准工序的优化方案。我们把  $X$  为某个特定值时，针对相应  $Y$  的数值进行的测量过程，称作一次测定。测定要付出一定的成本；定标误差也要折算成一定的成本。具体的成本函数如下：  
单点定标误差成本

$$s_{i,j} = \begin{cases} 0 & \text{if } |\hat{y}_{i,j} - y_{i,j}| \leq 0.5 \\ 0.5 & \text{if } 0.5 < |\hat{y}_{i,j} - y_{i,j}| \leq 1 \\ 1.5 & \text{if } 1 < |\hat{y}_{i,j} - y_{i,j}| \leq 2 \\ 6 & \text{if } 2 < |\hat{y}_{i,j} - y_{i,j}| \leq 3 \\ 12 & \text{if } 3 < |\hat{y}_{i,j} - y_{i,j}| \leq 5 \\ 25 & \text{if } |\hat{y}_{i,j} - y_{i,j}| > 5 \end{cases} \quad \text{对过大偏差的“惩罚性”}$$

其中，下标  $i$  代表样本序号 下标  $j$  代表观测点序号

对某一样本  $i$  的定标成本：  $S_i = \sum_{j=1}^{51} s_{i,j} + q \cdot n_i$  本课题， $q=12$ ， $n_i$  是该样本定标时所测定的

的点数 第一项：误差成本 第二项：测定成本

$$C = \frac{1}{M} \sum_{i=1}^M S_i$$

定标方案平均成本：

注：我们求取统计平均意义上的最好方法。总成本最低的方案，对特定样本的定标成本不一定最低。

## 2.2 数据样本分析

根据老师的教案以及我们的多次观测与实验，发现大多数样本都具有以下特点：

- (1) 呈现非线性。
- (2) 不同个体点，特性一致性不够强。
- (3) 根据大致的变化规律，可分为三部分。特性曲线按斜率变化大致可以区分为首段、中段、尾段三部分，中段的平均斜率小于首段和尾段。
- (4)  $Y$  取值随  $X$  取值的增大而单调递增。
- (5)  $X$  取值在  $[5.0, 10.0]$  区间内， $Y$  取值在  $[0, 100]$  区间内
- (6) 首段、中段、尾段单独都不是完全线性的，且不同个体的弯曲形态有随机性差异；
- (7) 不同个体的中段起点位置、终点位置有随机性差异。<sup>[1]</sup>

## 3. 选择拟合方式

### 3.1 了解拟合方式

数据样本实验数据的正确处理，关系到是否能达到实验目的，得出明确结论。传统的数据处理方法，很难得到一条很好地适应所有点的曲线，同时也无法估计所得曲线的精度，由此所确定的特征值就可能有较大的误差，且没有建立起由这些点构成曲线的数学模型，这些都将直接影响利用数学方法进行解析分析。通常的做法是在进行实验数据分析时，采用某种拟合方式，并在某种最佳准则下找出最合适的曲线。常见的拟合方式有基于最小二乘原理的多项式拟合、指数函数拟合、傅里叶级数拟合和插值计算等。由于这一部分仅仅讨论采取哪种拟合方式更好，所以可以先确定若干组特征点，然后做拟合。选择一种拟合（或插值）方法，也就是选择表达式

### 3.2 具体拟合方式的选择

查阅相关资料及翻看概率统计书籍，了解了相关知识后。主要可以采取的方法有如下两种：

一、插值

在特征点位置上，表达式与实验值无误差

二、拟合

在特征点位置上，表达式与实验值可以有误差（残差）

通过讲座我们了解到由于线性插值误差较大，而三次样条插值计算量较大，基于准确性和成本的考虑，应采用多项式拟合，由于不可能为线性，故从二次多项式开始考虑当阶数增加时，标准差降低，所得精度越来越高，但计算量也随之加大。除二阶拟合外，三阶四阶以至于更高阶相差不大。为了得到精度与成本的平衡，我们暂定选择三，四，五阶。结果见下面选择完特征点以后的测试。

## 4. 选择特征点

### 4.1 取点方案

即选择一种取点方案：位置、点数

注：在取点位置进行“单点测定”，单点测定的数值，都将作为拟合数据源  
因为根据老师上课讲的样本特性曲线按斜率变化大致可分为三段，因此我们先开始决定在三段上分别取点。每个样本有 51 个测定点，我们决定分别考虑取 6，7，，9 个点然后比较，分析。因为起点和重点肯定要取，所以就分别剩下 4，5，6 个点，然后分别分成 1，2，1；2，1，2；2，2，2 三种取点。后来，经过测试，发现平均取点更加方便，而且和选择取点理论上一样，所以才用平均取点，并且决定考虑取更多的点比较。

### 4.2 取点试验

下面是三，四，五次分别平均取 6，7，8，9 个点的结果

三次拟合：

取 6 个点

2     8     22     33     46     50

137.6919

取 7 个点

3     10     21     30     38     44     49

133.4264

取 8 个点

3     5     15     24     32     40     47     49

132.5416

取 9 个点

3     6     13     18     25     32     39     46     50

132.5714

取 10 个点

3     8     15     15     23     26     37     33     43     49

132.6162

四次拟合：

取 6 个点

3     12     23     33     43     50

117.2399

取 7 个点

3     10     16     24     33     43     50

115.7388

取 8 个点

3     4     10     21     30     39     48     50

115.9275

取 9 个点

3     2     10     19     27     34     40     48     50

114.9296

取 10 个点

2     8     15     18     24     29     35     39     50     46

114.4712

五次拟合：

取 6 个点

2     9     20     32     43     50

117.8582

取 7 个点

2     10     20     29     38     46     50

106.5885

取 8 个点

2     9     17     26     34     40     47     50

105.6343

取 9 个点

1     4     12     18     25     33     41     48     50

105.1087

取 10 个点

2     5     12     15     21     24     31     38     45     50

105.0554

### 4.3 取点分析

我们发现，随着取点的增加，成本分别在 8，7，10 个点左右趋于最小值然后增加. 并且拟合次数的增加最小成本减小。

其中，更改取点方案依靠退火算法

## 五. 采用模拟退火算法的求解过程

### 5.1 模拟退火算法（SA）介绍

寻找最优特征点的过程其实是一个搜索全局最优解的过程，如果使用暴力穷举法，加入取7个点，其运算次数将达到 $C_{51}^7 = 1.6$ 亿之多，严重浪费计算时间和空间。这里的最优解又是一个离散最优解的情况，使得求解更加困难。我们采用模拟退火算法最大程度的接近全局最优解。

### 5.2 模拟退火算法的大概操作步骤：

- (1) 设置初始温度  $t_0$ 、终止温度  $t_f$  及控制参数更新函数  $T(t)$ ；
- (2) 随机产生初始解  $x_0$ ，以此作为当前最优点  $x_{opt} = x_0$ ，计算目标函数值  $f(x_{opt})$ ；
- (3) 对当前最优点作一随机变动，产生一新解  $x_k$ ，计算新解的目标函数值  $f(x)$ ，并计算目标函数值增量  $\Delta f = f(x) - f(x_{opt})$ ；
- (4) 若  $\Delta < 0$ ，则接受该新解为当前最优点， $x_{opt} = x$ ；若  $\Delta \geq 0$ ，则以概率  $p$  的方式接

受该新解为当前最优点；

(5) 若  $k < L_k$ ，则  $k \leftarrow k + 1$ ，转(4)；

$t_k = T(t_{k-1})$ 。设置  $L_k$ ，令循环计数器初值  $k = 0$ ；

(6) 若  $t \geq T_f$ ，则转(3)；

若  $t < T_f$ ，则输出当前最优点，算法结束[2]

### 5.3 具体过程

我们将数据读入data矩阵，用best\_array矩阵记录当前最优选点方案。每次先从51个数据中随机选点，直到评估结果小于150，以此作为模拟退火的初始解。然后设置初始温度为100℃，退火速率为0.9进行模拟退火。每次对最优解中的每一个点做一次微调，分别为改选前一个点、改选后一个点或保持不变。调整后如果结果优于最优解则更新最优解，否则以p的概率接受当前解为最优解。在每次降温的时候输出当前温度和当前解，便于预估程序的运行时间。在最开始过程，为了检验算法的可行性，提运行效率，设置了起始温度为100度，结束温度为0.1度，降温系数为0.95

## 六. 结论

### 6.1 实验结果

最小成本：105

由于循环控制变量很大，我们简化了部分过程，每种多项式取固定点数大概只跑了10分钟，所以可以认定所得解不是该条件下的最优解，但已经能较好的反映问题，并且节约了时间成本。

### 6.2 实验结果分析

我们发现取点的不同最小成本随多项式的次数变化而变化，不过大概在7，8，9个点于左右达到最小。

### 6.3 对模拟退火算法的理解

- (1) 由于模拟退火法具有能够将大量数据进行局域化并局部计算的能力，其局部与整体的稳定性也确保了试验结果的准备性和高效性，因其得出的结果能够令人满意。
- (2) 必须要注意存储能力的增强，为避免搜索过程中由于执行概率接受环节而遗失当前遇到的最优解。

### 6.4 任务总结

首先，通过这门课，我们小组从接触到学会了数学建模、理论推算、算法制定等的综合应用，并对系统性能优化设计方案进行讨论和实现，且通过动手实验用数据证明数学模型及算法的有效性，受益匪浅。

其次，我们的综合能力也得到了训练。在此过程中，数学、科学和工程知识的综合应用能力，设计和进行实验以及对实验数据分析、整理的能力，发现、定义和解决实际工程问题的能力，团队协作和人际沟通能力，都得到了有效提高。

最后，感谢老师的悉心指导，感谢队友，感谢周围对我有帮助的同学！  
实现代码见后面。

## 七. 参考资料

- [1]袁炎.《统计推断讲座》讲义和参考资料;
- [2]百度文库《matlab 数据拟合实用教程》;

## 附录文本：实验代码

### 1.Cal\_cost

```
function re = Cal_cost(x_base, y_base, x_tmp, y_tmp)
grade = 0;
pol3 = polyfit(x_tmp,y_tmp,4);
for w = 1:51
    d = abs(polyval(pol3,x_base(w))-y_base(w));
    if d <= 0.5,
        grade = grade + 0;
    elseif d <= 1,
        grade = grade + 0.5;
    elseif d <= 2,
        grade = grade + 1.5;
    elseif d <= 3,
        grade = grade + 6;
    elseif d <= 5,
        grade = grade + 12;
    elseif d > 5,
        grade = grade + 25;
    end;
end;
grade = grade + 12 * 7;
if grade < 0, grade=0; end;
re = grade;
```

### 2. Main

```
data = xlsread('D:\data.xls');
best_array = zeros (1,7);
least_cost = 0;
sum_cost = 0;
T = 100;
cost = 1000;
while cost >= 150
    sum_cost = 0;
    ele1=1;
    ele2=randi([7,14]);
    ele3=randi([15,22]);
    ele4=randi([23,30]);
    ele5=randi([31,38]);
    ele6=randi([39,46]);
    ele7=51;
    random_array = [ele1, ele2, ele3, ele4, ele5, ele6, ele7];
```

```

for i = 1:469;
    tempdata=data([2*i-1,2*i],:);
    y_base=tempdata(2,:);
    x_base=tempdata(1,:);
    x_tmp=x_base(1,random_array);
    y_tmp=y_base(1,random_array);
    sum_cost=sum_cost+Cal_cost(x_base,y_base,x_tmp,y_tmp);
end
cost=sum_cost/469;
least_cost=cost;
best_array=random_array;
end
while T > 0.1
    T = 0.9 * T;
    sum_cost = 0;
    for i = 1:7
        while true
            b = random_array(i);
            c = randi(3);
            b = b + c - 2;
            if (b > 0 && b <= 51)
                break
            end
        end
        random_array(i) = b;
    end
    for i=1:469;
        tempdata=data([2*i-1,2*i],:);
        y_base=tempdata(2,:);
        x_base=tempdata(1,:);
        x_tmp=x_base(1,random_array);
        y_tmp=y_base(1,random_array);
        sum_cost=sum_cost+Cal_cost(x_base,y_base,x_tmp,y_tmp);
    end
    cost = sum_cost/469;
    e = exp((cost-sum_cost)/T*100);
    if (cost <= least_cost)
        least_cost = cost;
        best_array = random_array;
    elseif (rand > e)
        random_array=best_array;
    end
    disp(cost);
    disp(T);
end

```



```

end
disp(best_array);
disp(least_cost);

```

### 3. Main1

```

data = xlsread('D:\data.xls');
best_array = zeros (1,6);
least_cost = 0;
sum_cost = 0;
T = 100;
cost = 1000;
while cost >= 150
    sum_cost = 0;
    ele1=1;
    ele2=randi([2,14]);
    ele3=randi([15,27]);
    ele4=randi([28,40]);
    ele5=randi([41,50]);
    ele6=51;
    random_array = [ele1, ele2, ele3, ele4, ele5, ele6];
    for i = 1:469;
        tempdata=data([2*i-1,2*i],:);
        y_base=tempdata(2,:);
        x_base=tempdata(1,:);
        x_tmp=x_base(1,random_array);
        y_tmp=y_base(1,random_array);
        sum_cost=sum_cost+Cal_cost(x_base,y_base,x_tmp,y_tmp);
    end
    cost=sum_cost/469;
    least_cost=cost;
    best_array=random_array;
end
while T > 0.1
    T = 0.9 * T;
    sum_cost = 0;
    for i = 1:6
        while true
            b = random_array(i);
            c = randi(3);
            b = b + c - 2;
            if (b > 0 && b <= 51)
                break
            end
        end
    end
end

```

```

        random_array(i) = b;
    end
    for i=1:469;
        tempdata=data([2*i-1,2*i],:);
        y_base=tempdata(2,:);
        x_base=tempdata(1,:);
        x_tmp=x_base(1,random_array);
        y_tmp=y_base(1,random_array);
        sum_cost=sum_cost+Cal_cost(x_base,y_base,x_tmp,y_tmp);
    end
    cost = sum_cost/469;
    e = exp((cost-sum_cost)/T*100);
    if (cost <= least_cost)
        least_cost = cost;
        best_array = random_array;
    elseif (rand > e)
        random_array=best_array;
    end
    disp(cost);
    disp(T);
end
disp(best_array);
disp(least_cost);

```

#### 4. Main2

```

data = xlsread('D:\data.xls');
best_array = zeros (1,8);
least_cost = 0;
sum_cost = 0;
T = 100;
cost = 1000;
while cost >= 150
    sum_cost = 0;
    ele1=1;
    ele2=randi([2,9]);
    ele3=randi([10,18]);
    ele4=randi([19,27]);
    ele5=randi([28,36]);
    ele6=randi([37,45]);
    ele7=randi([46,50]);
    ele8=51;
    random_array = [ele1, ele2, ele3, ele4, ele5, ele6, ele7, ele8];
    for i = 1:469;
        tempdata=data([2*i-1,2*i],:);

```

```

        y_base=tempdata(2,:);
        x_base=tempdata(1,:);
        x_tmp=x_base(1,random_array);
        y_tmp=y_base(1,random_array);
        sum_cost=sum_cost+Cal_cost(x_base,y_base,x_tmp,y_tmp);
    end
    cost=sum_cost/469;
    least_cost=cost;
    best_array=random_array;
end
while T > 0.1
    T = 0.9 * T;
    sum_cost = 0;
    for i = 1:8
        while true
            b = random_array(i);
            c = randi(3);
            b = b + c - 2;
            if (b > 0 && b <= 51)
                break
            end
        end
        random_array(i) = b;
    end
    for i=1:469;
        tempdata=data([2*i-1,2*i],:);
        y_base=tempdata(2,:);
        x_base=tempdata(1,:);
        x_tmp=x_base(1,random_array);
        y_tmp=y_base(1,random_array);
        sum_cost=sum_cost+Cal_cost(x_base,y_base,x_tmp,y_tmp);
    end
    cost = sum_cost/469;
    e = exp((cost-sum_cost)/T*100);
    if (cost <= least_cost)
        least_cost = cost;
        best_array = random_array;
    elseif (rand > e)
        random_array=best_array;
    end
    disp(cost);
    disp(T);
end
disp(best_array);

```

```
disp(least_cost);
```

## 5. Main3

```
data = xlsread('D:\data.xls');
best_array = zeros (1,9);
least_cost = 0;
sum_cost = 0;
T = 100;
cost = 1000;
while cost >= 150
    sum_cost = 0;
    ele1=1;
    ele2=randi([2,8]);
    ele3=randi([9,15]);
    ele4=randi([16,22]);
    ele5=randi([23,29]);
    ele6=randi([30,36]);
    ele7=randi([37,43]);
    ele8=randi([44,50]);
    ele9=51;
    random_array = [ele1, ele2, ele3, ele4, ele5, ele6, ele7, ele8, ele9];
    for i = 1:469;
        tempdata=data([2*i-1,2*i],:);
        y_base=tempdata(2,:);
        x_base=tempdata(1,:);
        x_tmp=x_base(1,random_array);
        y_tmp=y_base(1,random_array);
        sum_cost=sum_cost+Cal_cost(x_base,y_base,x_tmp,y_tmp);
    end
    cost=sum_cost/469;
    least_cost=cost;
    best_array=random_array;
end
while T > 0.1
    T = 0.9 * T;
    sum_cost = 0;
    for i = 1:9
        while true
            b = random_array(i);
            c = randi(3);
            b = b + c - 2;
            if (b > 0 && b <= 51)
                break
            end
        end
    end
end
```

```

        end
    end
    random_array(i) = b;
end
for i=1:469;
    tempdata=data([2*i-1,2*i],:);
    y_base=tempdata(2,:);
    x_base=tempdata(1,:);
    x_tmp=x_base(1,random_array);
    y_tmp=y_base(1,random_array);
    sum_cost=sum_cost+Cal_cost(x_base,y_base,x_tmp,y_tmp);
end
cost = sum_cost/469;
e = exp((cost-sum_cost)/T*100);
if (cost <= least_cost)
    least_cost = cost;
    best_array = random_array;
elseif (rand > e)
    random_array=best_array;
end
disp(cost);
disp(T);
end
disp(best_array);
disp(least_cost);

```

## 6. Main4

```

data = xlsread('D:\data.xls');
best_array = zeros (1,10);
least_cost = 0;
sum_cost = 0;
T = 100;
cost = 1000;
while cost >= 150
    sum_cost = 0;
    ele1=1;
    ele2=randi([2,8]);
    ele3=randi([9,15]);
    ele4=randi([16,19]);
    ele5=randi([20,25]);
    ele6=randi([26,29]);
    ele7=randi([30,36]);
    ele8=randi([37,43]);
    ele9=randi([44,50]);

```

```

ele10=51;
random_array = [ele1, ele2, ele3, ele4, ele5, ele6, ele7, ele8, ele9, ele10];
for i = 1:469;
    tempdata=data([2*i-1,2*i],:);
    y_base=tempdata(2,:);
    x_base=tempdata(1,:);
    x_tmp=x_base(1,random_array);
    y_tmp=y_base(1,random_array);
    sum_cost=sum_cost+Cal_cost(x_base,y_base,x_tmp,y_tmp);
end
cost=sum_cost/469;
least_cost=cost;
best_array=random_array;
end
while T > 0.1
    T = 0.9 * T;
    sum_cost = 0;
    for i = 1:10
        while true
            b = random_array(i);
            c = randi(3);
            b = b + c - 2;
            if (b > 0 && b <= 51)
                break
            end
        end
        random_array(i) = b;
    end
    for i=1:469;
        tempdata=data([2*i-1,2*i],:);
        y_base=tempdata(2,:);
        x_base=tempdata(1,:);
        x_tmp=x_base(1,random_array);
        y_tmp=y_base(1,random_array);
        sum_cost=sum_cost+Cal_cost(x_base,y_base,x_tmp,y_tmp);
    end
    cost = sum_cost/469;
    e = exp((cost-sum_cost)/T*100);
    if (cost <= least_cost)
        least_cost = cost;
        best_array = random_array;
    elseif (rand > e)
        random_array=best_array;
    end
end

```

```
        disp(cost);  
        disp(T);  
end  
disp(best_array);  
disp(least_cost);
```