

统计推断在数模转换系统中的应用

组号：32 小组成员：侯碧潭 5140309070 黎灿 5140309161

摘要：本文是《统计推断在数模转换系统中的应用》的实验报告初稿。本文的研究对象是某些样品的某些数据，研究目的是为这些数据建立一个数学模型。本小组通过对样本取点，用 MATLAB 数学软件对其进行拟合求出其成本，再通过退火算法更换选点，进而比较求出最优方案。

关键字：MATLAB，拟合，插值，模拟退火算法，最优解

1 引言

在工程应用与实际生产中我们常常遇到一些定标问题，我们需要找到两个变量之间的关系来更好地了解其内在联系，我们通过抽样取点并将之拟合成曲线从而确定其关系。但在实际生产过程中，测量大量的样本会花费大量的时间与成本，不利于生产，于是我们需要取出既不失精确性，又能极好的控制成本的点来拟合出变量之间的关系。我们在本课题中通过模拟退火算法来达到这一目的。

2 课题实际背景[1]

2.1 课题概述

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）。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。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。

2.2 课题模型

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。

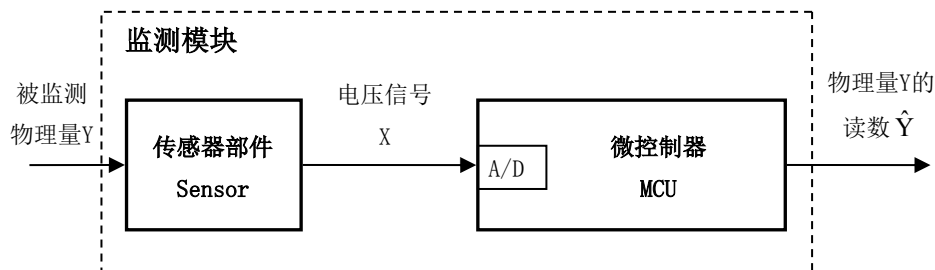


图 1 监测模块组成框图

监测模块的组成框图如图 1。其中，传感器部件（包含传感器元件及必要的放大电路、调理电路等）的特性是我们关注的重点。传感器部件监测的对象物理量以符号 Y 表示；传感部件的输出电压信号用符号 X 表示，该电压经模数转换器（ADC）成为数字编码，并能被微

处理器程序所读取和处理，获得信号 \hat{Y} 作为 Y 的读数（监测模块对 Y 的估测值）。

所谓传感特性校准，就是针对某一特定传感部件个体，通过有限次测定，估计其 Y 值与 x 值间一一对应的特性关系的过程。数学上可认为是确定适用于该个体的估测函数 $\hat{y} = f(x)$

的过程，其中 x 是 x 的取值， \hat{y} 是对应 Y 的估测值。

考虑实际工程中该监测模块的应用需求，同时为便于在本课题中开展讨论，我们将问题限于 x 为离散取值的情况，规定

$$X \in \{x_1, x_2, x_3, \dots, x_{50}, x_{51}\} = \{5.0, 5.1, 5.2, \dots, 9.9, 10.0\}$$

相应的 Y 估测值记为 $\hat{y}_i = f(x_i)$ ， Y 实测值记为 y_i ， $i = 1, 2, 3, \dots, 50, 51$ 。

2.3 成本计算

为评估和比较不同的校准方案，特制定以下成本计算规则。

- 单点定标误差成本

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

单点定标误差的成本按式（1）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

$$S_i = \sum_{j=1}^{51} s_{i,j} + q \cdot n_i \quad (2)$$

对样本 i 总的定标成本按式（2）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）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。

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

总成本较低的校准方案，认定为较优方案。

3 设计思路

3.1 基本思路

本小组通过对样本数据取点，用 MATLAB 数学软件对其进行拟合成函数并求出其成本，再通过退火算法不断更换选点，求出不同方案的成本，通过比较各个方案的成本从而得到最优解。

3.2 取点思路

一开始，我们首先从 400 组数据中随机选出 100 组，每组随机选取 6 个点，通过对这 6 个点拟合可得到两个变量之间的关系，再用 400 组实际数据与拟合曲线上的点求出其 400 组的平均成本，再通过模拟退火算法变换选点，再次求出成本，不断重复通过比较成本最后可得到较优的一组点。

经过后续多次测试 5 个点、6 个点、7 个点，发现选取 5 个点成本更低，最终决定选取 5 个点。

3.3 模拟思路

关于拟合，比较流行的拟合方法有三次样条差值方法。它属于分段差值。“三次”是指函数的次数，“样条”源于过去绘图员使用的一种绘图工具样条，它是用富于弹性、能弯曲的木条（或塑料）制成的软尺，把它弯折靠近所有基点并用画笔沿着样条就可以画出连续基点的光滑曲线。三次样条差值用分段的多项式构造一个整体上具有函数、一阶和二阶导连续的函数，近似的替代已知函数 $f(x)$ 。

关于函数插值的 MATLAB 的实现，命令调用的格式 $Y0=interp1(x,y,x0,'method')$ ，其中 `method` 有以下 4 种参数可选。`nearest`：最近插值；`linear`：线性插值；`pchip`：分段三次插值；`spline`：三次样条插值。其中 `spline` 为老师提供的拟合方法。经过 MATLAB 代码实现后，发现，在该问题中 `pchip` 拟合方法更适合我们的课题，所计算的成本也最低，故最终采用此种方法。[2]

在最终决定拟合方法之前，本小组也尝试过使用数据的多项式曲线拟合。所谓曲线拟合，又称函数逼近，是求近似函数的又一类数值方法。他不要求近似函数再节点处与函数同值，即不要求近似函数过已知点，只要求它尽可能的反映给定数据点的基本趋势，在某种意义上与函数的逼近。在 MATLAB 中，与三次样条差值一样，也有自己的命令的调用格式，所以使用很方便。但是，也许由于该拟合方法拟合的曲线过于精确，且不要求经过样本点，所以运行的时间复杂度过高，即程序运行时间太长，且最终的结果并没有更加优秀多少。故最终放弃此法。

3.4 模拟退火算法[3]

3.4.1 原理

由初始解 i 和控制参数初值 t 开始，对当前解重复“产生新解→计算目标函数差→接受或舍弃”的迭代，并逐步衰减 t 值，算法终止时的当前解即为所得近似最优解。

3.4.2 模型建立

模拟退火的基本思想：

(1) 初始化: 初始温度 T (充分大), 初始解状态 S (是算法迭代的起点), 每个 T 值的迭代次数 L

(2) 对 $k=1, \dots, L$ 做第(3)至第 6 步:

(3) 产生新解 S'

(4) 计算增量 $\Delta t' = C(S') - C(S)$, 其中 $C(S)$ 为评价函数

(5) 若 $\Delta t' < 0$ 则接受 S' 作为新的当前解, 否则以概率 $\exp(-\Delta t' / T)$ 接受 S' 作为新的当前解。

(6) 如果满足终止条件则输出当前解作为最优解, 结束程序。

终止条件通常取为连续若干个新解都没有被接受时终止算法。

(7) T 逐渐减少, 且 $T > 0$, 然后转第 2 步。

3.4.3 算法步骤

模拟退火算法新解的产生和接受可分为如下四个步骤:

第一步是由一个产生函数从当前解产生一个位于解空间的新解; 为便于后续的计算和接受, 减少算法耗时, 通常选择由当前新解经过简单地变换即可产生新解的方法, 如对构成新解的全部或部分元素进行置换、互换等, 注意到产生新解的变换方法决定了当前新解的邻域结构, 因而对冷却进度表的选取有一定的影响。

第二步是计算与新解所对应的目标函数差。因为目标函数差仅由变换部分产生, 所以目标函数差的计算最好按增量计算。事实表明, 对大多数应用而言, 这是计算目标函数差的最快方法。

第三步是判断新解是否被接受, 判断的依据是一个接受准则, 最常用的接受准则是 Metropolis 准则: 若 $\Delta t' < 0$ 则接受 S' 作为新的当前解 S , 否则以概率 $\exp(-\Delta t' / T)$ 接受 S' 作为新的当前解 S 。

第四步是当新解被确定接受时, 用新解代替当前解, 这只需将当前解中对应于产生新解时的变换部分予以实现, 同时修正目标函数值即可。此时, 当前解实现了一次迭代。可在此基础上开始下一轮试验。而当新解被判定为舍弃时, 则在原当前解的基础上继续下一轮试验。

3.5 关于模拟退火算法的探究

3.5.1 初稿存在的疑虑

(1) 对于模拟退火算法中退火常数以及初温的确定存在疑虑

(2) 对于模拟退火算法中学习因子不知道具体代表什么含义

(3) 对于模拟退火算法中以一定概率接受新解存在疑虑

3.5.2 终稿的解决方案

(1) 经过与袁炎老师的面谈以及查阅资料, 我们明白了退火常数以及初温与退火过程并没有直接联系, 退火常数与初温决定的是退火过程的快慢以及何时结束。退火过程越慢, 得到的答案越精确, 但电脑需要运行的时间越久; 退火过程越快, 电脑需要运行的时间短, 但得到的答案较粗糙。本小组考虑到这两方面的影响, 决定采用初温为 2000°C (1000°C 、 3000°C 、 5000°C 均可, 视所需的精确度而定), 终温为 1°C , 退火常数为 1, 每运行一次, 温度减 1, 当最终温度小于 1°C 时, 循环结束。

(2) 经过与袁炎老师的面谈以及小组的讨论, 我们明白了学习因子只是模拟退火算法中交换选点时所需的常数, 可自行确定。经本小组的讨论, 本小组决定摒弃资料书中所提供的换点方法, 自己设计了微调与大幅度调整两种方案, 微调针对的是新解更优秀, 每次随机变换一个选点; 大幅度调整针对的是新解不比原来的解优秀, 但以一定概率接受其存在, 每次随机变换 3 个选点。

(3) 经过与袁炎老师的面谈以及小组的激烈讨论, 我们明白了以一定概率接受新解的意义是为了排除局部最优解、求得全局最优解。但本小组在如何体现这一概率以及对应情况该如

何换点上存在分歧，最后经过激烈的讨论与演算，本小组决定微调与大幅度调整两种方案：若新解比原来的优秀，则接受，并微调换点，即每次随机更换一个选点；若新解没有原来的优秀，但以一定概率接受其存在，并大幅度调整，即每次随机更换三个选点；若新解没有原来的优秀，并且在概率上也未接受，则接受原来的解，并微调。

4 算法思路及程序解释（核心部分）

首先，解的初始值为[1,2,3,4,5]，只不过是随机取的一组值，可以任意选取，并不影响最终结果。

关于初温 T ，只不过是决定何时达到循环条件退出循环，而退火常数 k ，是决定退火的快慢。每进入一次循环，就进行一次降温，我们设终止温度为 1 度。

变量 $aFinal$ 和 $pBest$ 是最终的取点和最终的最优解。这里本身有一个 $pNew$ 就足够了，但是，为了防止当以概率接受较差的解的时候，而恰恰这就是最优解，以致错过该最优解，经过本小组讨论，故采用 $pBest$ 进行一种的类似记忆的功效。

由于进入循环后要条件判断，而第一个新解对于最终的最优解来说是无关紧要的，为了方便程序编写，而又不影响性能，故采用第一个新解 $pNew=pOld+1$ 恒大于第一个随机解，进入循环后直接进入第二个条件判断，进行大幅调整，算法正式开始。

在循环中，当新解较差时，不能直接接受，要以一定的概率接受。先有一个 0~1 之间的随机数，与 Q 比较，若 Q 较大，则接受，其中 $Q=\exp(-(pNew-pOld)/T)$ ，也就是说新解比旧解大的越多，温度越低， Q 越小，则更难以接受。

接受较差解的时候，则进行“大幅度调整”。这是由于此解较差，样本值也许偏离最优解远一些，故要进行大调整，以使其向最优解逼近。关于最优解，下文会再详细叙述。

在循环中，当新解比旧解更加优秀时，则毫不犹豫接受它。但不能直接把 $pNew$ 赋值给 $pBest$ ，因为当你之前以概率接受过一个较差的解的时候，这里的 $pNew$ 可能比这个较差解要优秀，但比最早记忆的 $pBest$ 要差一些，那么一旦赋值，则 $pBest$ 不是整个过程中出现的最优解，这就失去了记忆的作用。

当接受一个更优秀的解的时候，说明这里的样本值在趋近于最优的值，故不宜进行大幅度调整，导致产生的新样本又偏离的最优解。故采用微调 $small_change$ 进行调整。关于微调，下文会再详细叙述。

微调：当新解优秀时，则微调样本，即只修改一个值，以引导其趋向最优解。调整的方法为：先在 1~5 之间产生一个随机数，以此来确定修改样本的第几个值，然后在 1-51 之间随机产生一个数，替代原来相应位置的样本值。但是，这样做有一个问题，如果替换一个值以后，新得到的值与其余的四个有可能相等，这样的话，会导致这样的新解的成本必然很大，与最优解的差距只会越来越大，并不能对最优解的产生有什么帮助，因此，在微调的程序中，我用一个 `while` 循环来判断样本中是否有两个值相等，如果有相等情况，重复进行上述同样的操作，直到五个值两两不相等，微调结束。

大幅度调整：当以概率接受一个较差的解的时候，我们采用大幅度调整的方法，即变换三个点。做法是这样的：先随机产生三个 0~5 之间的数，以此来决定变换样本中的哪三个点。然后在 1~51 之间产生随机数，来替换样本中相应位置的的值。但是，如前面微调所述，会出现一个样本五个点中存在相等的情况，这不是我们期望的，所以，和上面的微调一样，也通过 `while` 循环进行判断，直到两两不相等，大幅度调整结束。

当循环结束时，前一次循环中产生的 $pNew$ 没有用到，故在最后输出结果之前，先将其与 $pBest$ 进行比较，选择最优秀的作为 $pBest$ ，最终输出。

5 结论

多次运行程序后, 得到一组最优的解如下:

```
aFinal = [ 16  50  27  37  4]
```

pBest = 86.5760.

6 参考文献

- [1]上海交通大学电子系. 统计推断在数模转换系统中的应用课程讲义
- [2]张德丰等 主编 《MATLAB 数值计算方法》 机械工业出版社
- [3]百度百科. 模拟退火算法
- [4]龚纯, 王正林 编著 《精通 MATLAB 最优化计算》第 2 版
- [5]上届实验报告. 第 47 组
- [6]孙祥 徐流美 吴清编著 《MATLAB7.0 基础教程》 清华大学出版社
- [7]周开利 邓春晖 主编 《MATLAB 基础及其应用教程》 北京大学出版社

7 致谢

非常感谢袁炎老师，袁炎老师在课堂上的讲解以及后来的面谈解决了我们许多困惑，让我们在以前学长学姐的报告基础上有了自己想法；同时在面谈时为我们细心讲解了模拟退火算法的疑惑，让我们更深入地理解了退火算法，才使得我们有能力自己制定换点方案。

同时通过这门课程，我们学会了如何使用 MATLAB，也进一步了解了数学建模的思想方法，这门课程让我获益颇深。

最后，仍然感谢袁炎老师以及助教老师的帮助。本小组的这篇论文仍有许多不足，恳请老师批评与指正！

8 附录

8.1 代码展示

(1) 主函数

```
%主程序
```

```
a=[1,2,3,4,5];
```

```
%初始值
```

```
aFinal=a;
```

T=2000;

初温 可根据精细度不同,选取 1000,3000,5000 等不同的初温

 $k=1;$

%退火常数

```
p0=test ur answer(a);
```

%初始成本

```

pBest=p0; %记忆功能，以防错过最优解
pOld=p0;
pNew=pOld+1; %这里有话要说!!!
while (T>1)
    T=T-1;
    if (pNew<pOld) %若新解更加优秀，则接收
        pOld=pNew;
        if pBest>pNew %这步不可缺少!!!
            pBest=pNew;
            aFinal=a;
        end
        a=small_change(a);

    else % pNew>=pOld %若新解较之原来的的更差，则以概率接受
        Q=exp((pOld-pNew)/T); %分子分母需要进一步说明
        if rand(1,1)<=Q %Q 和 0~1 之间的随机数比较，以此选择是否接受
            pOld=pNew;
            a=big_change(a);
        end
    end
    pNew = test_ur_answer(a);
end

if (pNew<pBest)
    pBest=pNew;
    aFinal=a;
end

fprintf('\n 经计算，本次程序运行最优解如下： \n');
aFinal
pBest

```

(2) 成本函数

% 成本函数

```
function cost=test_ur_answer(a)
```

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

```

(3) 定标函数

```

% 定标计算方法
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');
```

```
%y1=interp1(x_premea,y0_premea,x,'spline');%该拟合方法为老师所提供，为三次样条差值
```

```
%A=polyfit(x_premea,y0_premea,3);%此两行该拟合方法为多项式拟合，但运行时间太长，尽管可能更精确
```

```
%y1=polyval(A,x); %且最终的结果并不会优化多少，故最终放弃此法
```

End

(4) 微调函数

%微调

```
function a=small_change(a) %a 为一行五列
```

```
num=randperm(5,1);
```

```
a(num)=randperm(51,1);
```

```
%while(a(1)==a(2)||a(1)==a(3)||a(1)==a(4)||...该注释为起初采用六点拟合时的 while 循环
```

```
% a(1)==a(5)||a(1)==a(6)||a(2)==a(3)||...
```

```
% a(2)==a(4)||a(2)==a(5)||a(2)==a(6)||...
```

```
% a(3)==a(4)||a(3)==a(5)||a(3)==a(6)||...
```

```
% a(4)==a(5)||a(4)==a(6)||a(5)==a(6))
```

```
while(a(1)==a(2)||a(1)==a(3)||a(1)==a(4)||...
```

```
 a(1)==a(5)||a(2)==a(3)||...
```

```
 a(2)==a(4)||a(2)==a(5)||...
```

```
 a(3)==a(4)||a(3)==a(5)||...
```

```
 a(4)==a(5))
```

```
 num=randperm(5,1);
```

```
 a(num)=randperm(51,1);
```

end

(5) 大幅调整函数

%大幅调整

```
function a=big_change(a) %a 为一行五列
```

```
num=randperm(5,3);
```

```
a(num(1))=randperm(51,1);
```

```
a(num(2))=randperm(51,1);  
a(num(3))=randperm(51,1);
```

%while(a(1)==a(2)||a(1)==a(3)||a(1)==a(4)||... 该注释为起初采用六点拟合时的 while 循环

```
%      a(1)==a(5)||a(1)==a(6)||a(2)==a(3)||...  
%      a(2)==a(4)||a(2)==a(5)||a(2)==a(6)||...  
%      a(3)==a(4)||a(3)==a(5)||a(3)==a(6)||...  
%      a(4)==a(5)||a(4)==a(6)||a(5)==a(6))  
while(a(1)==a(2)||a(1)==a(3)||a(1)==a(4)||...  
      a(1)==a(5)||a(2)==a(3)||...  
      a(2)==a(4)||a(2)==a(5)||...  
      a(3)==a(4)||a(3)==a(5)||...  
      a(4)==a(5))  
  
      %num=randperm(5,3);  
      a(num(1))=randperm(51,1);  
      a(num(2))=randperm(51,1);  
      a(num(3))=randperm(51,1);  
end
```