统计推断在数模转换系统中的应用

组号 44 组　　刘和 5140309042 上官仲恺 5140309328

摘要：本课程主要展示了统计推断在数模，模数转换系统中的应用。以某电子产品的内部传感器部件的输入输出特性为研究对象，运用一定的数理统计方法，将实验室测得的四百组数据通过特征点选取、算法研究、拟合比较等一系列过程，借助 MATLAB 矩阵实验室工具来建立数学函数模型，最终达到以少量数据反映整体系统特性的效果，从而根据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。

关键词：统计推断，曲线拟合，模拟退火算法，成本，MATLAB

Application of Statistical Inference in AD&DA Inverting System

Group Number: 44　　He Liu 5140309042 ZhongkaiShangGuan 5140309328

ABSTRACT: This course in order to demonstrate the application of statistical inference in AD&DA inverting system. Using the output and input characteristics of a sensor inserted in some electronic product as the object of study, utilizing certain mathematical statistics methods, selecting the feature points in four hundred data measured from the laboratory, doing research in the algorithm, comparing a series of fitting modes, and establishing the mathematical function model with the help of MATLAB, we finally made the effect that a few data can show the identity of whole system come true, and designed a reasonable sensor-calibration program with logical cost according the volume production of module.

Key words: Statistical Inference, Curve Fitting, Simulated Annealing, Cost, MATLAB

参考他人报告或代码的申明

统计推断课程，2015 年秋季学期第 44 组，成员 刘和 学号 5140309042, 上官仲恺 学号 5140309328, 在报告编写过程中，以下方面参考了往届报告，现列表说明：

表 1 参考申明

主要参考项目	说明
代码方面	《统计推断在数模模数转换中的应用》，谢昊男，2013 年秋季学期，组号 06 参照了此代码的思路。
算法描述方面，包含流程图	《统计推断在数模转换系统中的应用》，肖弼，2013 年秋季学期，组号 48 引用了其模拟退火算法流程图以及拟合方法对比图。

1 引言

在工程上，产品往往被要求达到一定的精度，为此需对生产出的产品进行测定，本报告中传感器部件是一种典型的需要多次测量的产品，部分的模拟量函数关系可以通过数学关系

直接推导出来，但实际电路的误差、噪声等影响较大，通过数学推导并不可靠。因此研究实测的一组数据之间的函数关系，能够比较准确地反映出输出和输入之间的关系。而对于工业大规模批量生产，减少测定的数量即可以节约大量生产成本。因此探究如何选定尽可能少的点达到推定整体曲线的误差尽可能小是有现实意义的。

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。[1] 检测的工序一方面要保证精度，另一方面要考虑到控制的操作成本，因此我们需要通过研究该系统特性来找出一种方法来衡量产品的品质。本次研究中， 我们通过自行讨论选取的算法以及模拟退火算法的对比来进行四百组数据的处理，结合 MATLAB 工具来建立数学函数模型，并找到成本相对最小的传感特性校准（定标工序）方案。

2 具体问题

2.1 数学模型

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。

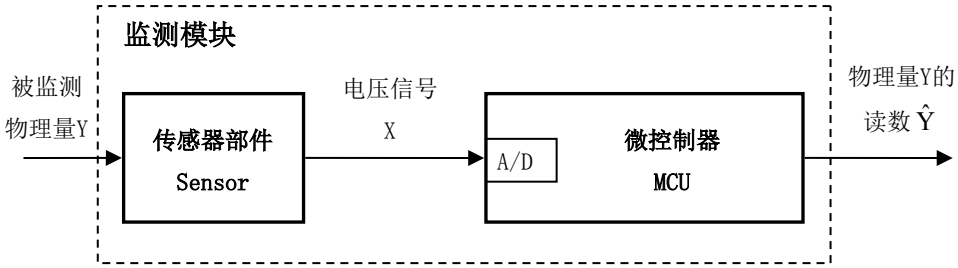

图 1 监测模块组成框图

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

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

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

2.2 传感器特性

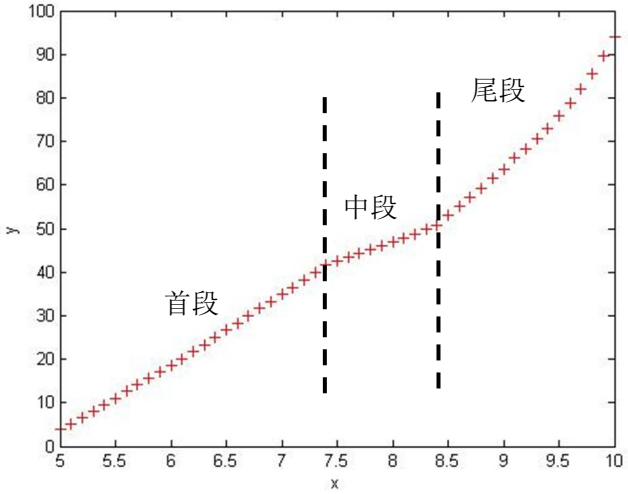

图 2 传感特性图示

一个传感部件个体的输入输出特性大致如图 2 所示，有以下主要特征：

(1）Y 取值随 X 取值的增大而单调递增；

(2）X 取值在[5.0,10.0]区间内，Y 取值在[0,100]区间内；

(3）不同个体的特性曲线形态相似但两两相异；

(4）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；

(5）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；

(6）不同个体的中段起点位置、终点位置有随机性差异。

2.3 成本计算规定

为评估和比较不同的校准方案，特制定以下成本计算规则。

（1）单点定标误差成本

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

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。

（2）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。

（3）某一样本个体的定标成本

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

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。

（4）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。

$$C = \frac{1}{M} \sum_{i=1}^{M} S_i \qquad (2\text{-}3)$$

总成本较低的校准方案，认定为较优方案。

3 算法设计实现与描述

3.1 自选算法描述与拟合方法

由于在进行数据处理前，我们并未对以模拟退火算法为代表的启发式搜索进行了解，经过讨论我们得出了一种特别粗糙且不成熟的拟合算法方式，并于此进行描述以便与模拟退火算法进行比较：

首先，对于每组 X 的值对应了 400 组数据，我们决定人为地将其分为 4 份，每份 100 组数据，再在这 4 份中每份我们通过每相隔 25 组数据选取 4 个数据点，做出四个散点图，如图 3 图 4 图 5 图 6 所示；

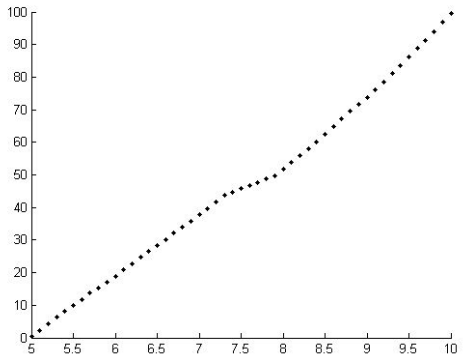

图 3 散点图（1）

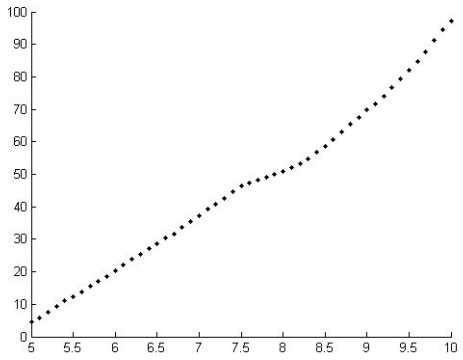

图 4 散点图（2）

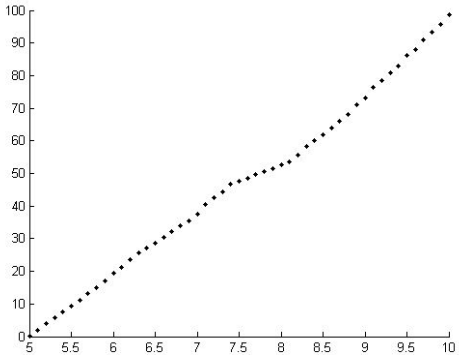

图 5 散点图（3）

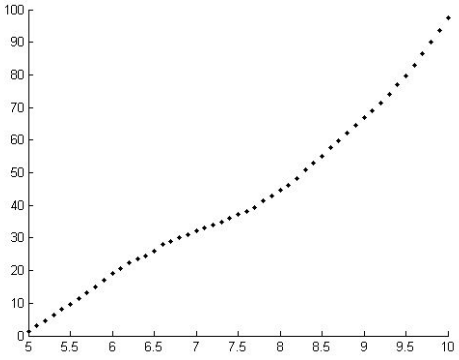

图 6 散点图（4）

由散点图可以看出来曲线大致三段的区间范围，经过讨论分析，我们选取的是 X 范围 1~25 为上段，26~35 为中段，36~51 为下端，分别进行三段的拟合。

接下来，我们采用多项式拟合的方法进行曲线拟合，根据初稿的对实验数据的拟合，多项式拟合出的多项式曲线更具有拟合的准确性。在本实验中采用最高次项为三次的多项式拟合。 即设一在区间[a,b]上的 n 阶多项式函数

$$P(x)=\sum_{k=0}^{n}a_k x^k \tag{3-1}$$

其中$a = x_0 < x_1 < \cdots < x_n = b$是数据点，则它为区间[a,b]上对数据点拟合的多项式。由定义可看出，它有 n+1 个待定系数。我们利用 MATLAB 中已有的函数

$$a = ployfit(x_0, y_0, m) \tag{3-2}$$

其中输入参数 x0，y0 为要拟合的数据，m 为拟合多项式的次数，输出参数 a 为拟合多项式系数。多项式在 x 处的值 y 可用 MATLAB 中的 y=polyval(a,x)函数进行计算。多项式拟合的次数在一定范围内越高，方差等参数越小，拟合曲线与实际测量点的相关性越高，拟合程度越好。但次数的增高导致算法运行时间的延长，效率的降低，而且当次数超过一定范围时，甚至适得其反。例如用 5，6 次函数拟合，运行时间是很难让人接受的，所得结果误差反而越来越大，可见不一定次数越高，拟合曲线就越接近实际曲线。反复测试后，3 次曲线拟合的效果最好。

接下来是计算定标成本，我们将拟合好的多项式参数储存在函数体中，并将 400 组数据一一代入进行检验，这个过程是十分繁琐的，我们也没有想到怎样更好的办法进行解决，最后得出的校准方案总成本 C 约为 641，通过对比上一届学长运用模拟退火算法或者模拟退火算法得出的校准方案总成本，我们发现这是异常大的，可以说是根本没有减少什么效率，所以我们这次拟合方案的尝试的结果不容乐观，不过我们通过这个过程了解了统计推断以及拟合的概念，同时意识到了症结所在以及启发式搜索算法的意义所在——虽然我们利用函数 plotfit 进行拟合的曲线比较好，但是采用了 51 个数据点，而且不是在 400 组数据中进行整体分析而是在这 51 个数据点中进行拟合的，没有最优的选择思想，因此成本异常高，我们进行了深刻的反思，觉得此次尝试与错误为我们进行以下通过模拟退火算法来确定校准方案奠定了基础。

其实在本课题中，选择测定点的组合方案，相当于解一个组合优化问题。备选的组合方案数量巨大，是典型的 NP-hard 问题（即算法复杂度不能用问题阶数 n 的多项式来表示）。例如，不妨假设最优方案是选取 9 个测试点，则总共的选择方案有约为 30 亿种。对于如此巨大的计算量，目前的计算机尚难以用暴力穷举的方法进行求解。对于 NP-hard 问题，可以尝试使用启发式搜索算法来求解：典型的启发式搜索有模拟退火算法等。

以上过程实现代码将附在附录中。

3.2 模拟退火算法的基本原理

“模拟退火”的原理和金属退火的原理近似：我们将热力学的理论套用到统计学上，将搜寻空间内每一点想像成空气内的分子；分子的能量，就是它本身的动能；而搜寻空间内的每一点，也像空气分子一样带有“能量”，以表示该点对命题的合适程度。算法先以搜寻空间内一个任意点作起始：每一步先选择一个“邻居”，然后再计算从现有位置到达“邻居”的概率。模拟退火流程图如图所示：

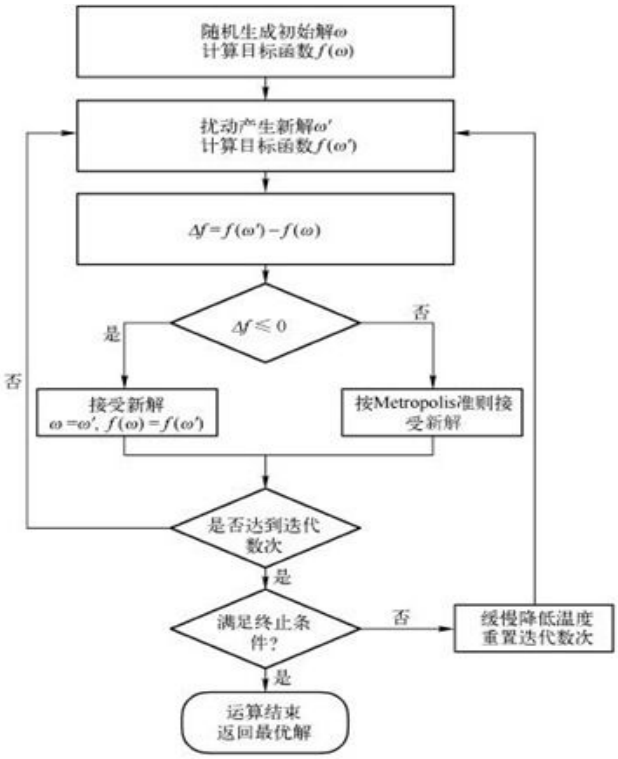

图 7 模拟退火算法流程图

模拟退火算法可以分解为解空间、目标函数和初始解三部分。

(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，……，L 做第(3)至第 6 步：

(3) 产生新解 S′；

(4) 计算增量Δt′=C(S′)-C(S)，其中 C(S)为评价函数；

(5) 若Δt′<0 则接受 S′作为新的当前解，否则以概率 exp(-Δt′/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；

(7) T 逐渐减少，且 T->末温度，然后转第 2 步。

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。[2]

模拟退火算法是一种随机算法，并不一定能找到全局的最优解，可以比较快的找到问题的近似最优解。 如果参数设置得当，模拟退火算法搜索效率比穷举法要高。

3.3 模拟退火算法的实现

(1) 首先确定测试点的个数的大概范围，然后随机产生测试点为初始状态。在本实验中，测试点的个数分别为 7,6,5,4。

(2) 每个状态的能量即为所有样本定标成本的平均值。初始温度设为 T=100，末温度设为 T_min=0.01，降温比例设为 r=0.97，经测试，T_min 大约等于 T 乘 r 的 303 次方，就是说总共测试了 303 组数据，num 最终为 303，已经可以充分反映出 400 组测试点的规律了。随着温度的降低，能量会越来越低。典型的模拟退火算法对于每个温度，内部循环 num 次取较优解，称为蒙特卡洛循环。状态更新方法为对当前状态进行随机扰动，即随机改变某个测试点的位置，计算新状态的能量，然后选择是否接受新状态。

(3) 模拟退火算法的伪代码：

```
while( T > T_min )
{
    dE = J( Y(i+1) ) - J( Y(i) ) ;

    if ( dE >=0 ) //表达移动后得到更优解，则总是接受移动
Y(i+1) = Y(i) ; //接受从 Y(i)到 Y(i+1)的移动
    else
    {
// 函数 exp( dE/T )的取值范围是(0,1) ，dE/T 越大，则 exp( dE/T )也
if ( exp( dE/T ) > random( 0 , 1 ) )
Y(i+1) = Y(i) ; //接受从 Y(i)到 Y(i+1)的移动
    }
    T = r * T ; //降温退火 ，0<r<1 。r 越大，降温越慢；r 越小，降温越快
    /*
    * 若 r 过大，则搜索到全局最优解的可能会较高，但搜索的过程也就较长。若 r 过小，则搜索的过程会很快，但最终可能会达到一个局部最优值
    */
    i ++ ;
}
```

(4) 对于模拟退火算法伪代码的说明：

J(y)：在状态 y 时的评价函数值

Y(i)：表示当前状态

Y(i+1)：表示新的状态

r：用于控制降温的快慢

T：系统的温度，系统初始应该要处于一个高温的状态

T_min：温度的下限，若温度 T 达到 T_min，则停止搜索

若 J(Y(i+1))<= J(Y(i))(即移动后得到更优解)，则总是接受该移动。若 J(Y(i+1))>J(Y(i))(即移动后的解比当前解要差)，则以一定的概率接受移动，而且这个概率随着时间推移逐渐降低（逐渐降低才能趋向稳定）。这里的“一定的概率”的计算参考了金属冶炼的退火过程，这也是模拟退火算法名称的由来。根据热力学的原理，在温度为 T 时，出现能量差为 dE 的降温的概率为 P(dE)，表示为：

$$P(dE) = \exp(dE/(kT)) \quad (3\text{-}3)$$

其中 k 是一个常数，在本实验中我们取 k=0.97，exp 表示自然指数，且 dE<0。这条公式说白了就是：温度越高，出现一次能量差为 dE 的降温的概率就越大；温度越低，则出现降温的概率就越小。又由于 dE 总是小于 0（否则就不叫退火了），因此 dE/kT < 0 ，所以 P(dE)的函数取值范围是(0,1) 。随着温度 T 的降低，P(dE)会逐渐降低。我们将一次向较差解的移动看做一次温度跳变过程，以概率 P(dE)来接受这样的移动。[3]

(5) 以下为取七个点时的模拟退火算法最优成本收敛曲线：

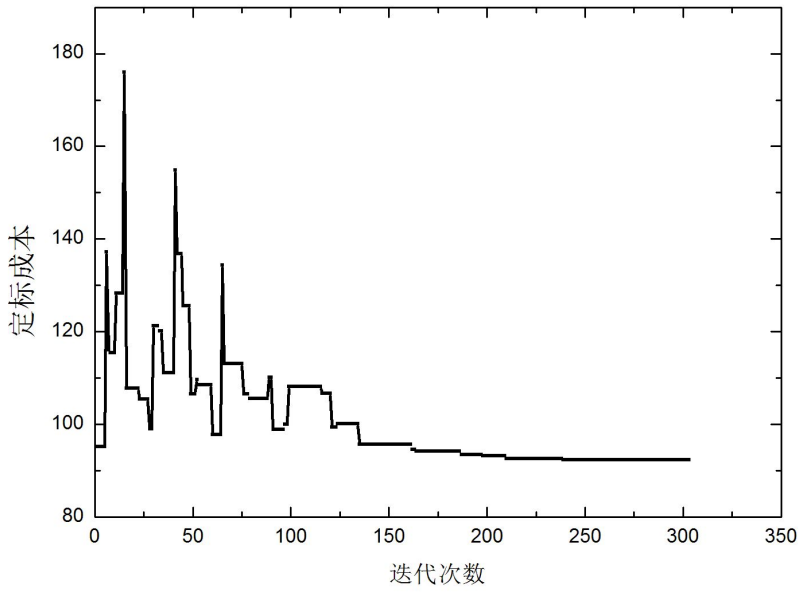

图 8 本实验中模拟退火算法成本收敛曲线

3.4 定标拟合方法的选取

有两种拟合方式可供选取：三次样条插值 spline 和多项式插值 pchip。spline 提供的函数 s(x)的构建方法和 pchip 里面的函数 p(x)完全相同，只不过在 X(j)处的斜率的选择方法不一样，spline 函数的 s(x)在 X(j)的二阶导数 D^2s(x)也是连续的，这导致了如下结果：spline 更加光滑，也就是说，D^2s(x)是连续的。如果数据是一个光滑函数的值，则 spline 更加精确。如果数据不是光滑的，则 pchip 不会超过目标值，也不太震荡。pchip 建立的难度较小。spline 比 pchip 光滑，样条的两阶导数连续，而 pchip 一阶导数连续。不连续的两阶导数隐含着不连续的曲率。人的眼睛可以检测出图形上曲率的不连续。另一方面，pchip 是保形状的，而 spline 不一定保形状。[4]

综合上述以及实践对比，我们发现 spline 计算的成本较低，故更加适合，于是选择了三次样条插值法进行拟合曲线。

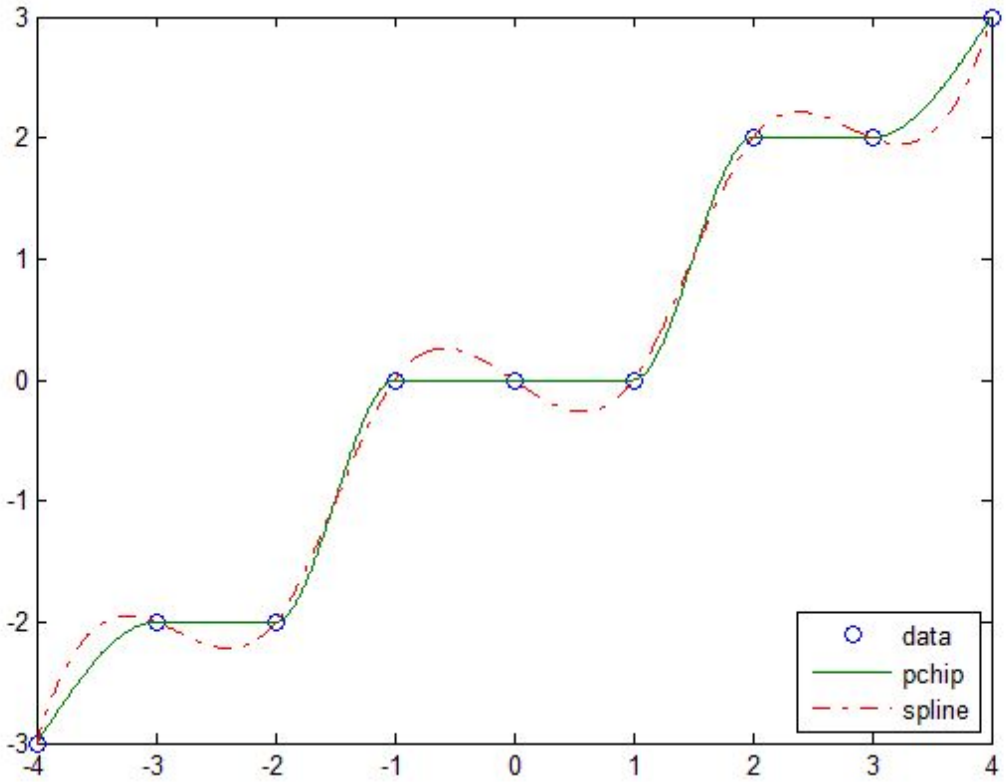

图 9 三次样条插值法 spline 和多项式插值法的区别图

4 运算结果分析与对比

多项式插值拟合法：

最优取点方案[3, 13, 20, 26, 32, 39, 49] 成本：92.31

最优取点方案[3, 13, 21, 27, 36, 48] 成本：88.13

最优取点方案[4, 15, 25, 35, 48] 成本：85.15

最优取点方案[4, 21, 36, 48] 成本：103.44

三次样条插值法：

最优取点方案[2, 11, 20, 29, 35, 45, 51] 成本：94.81

最优取点方案[3, 15, 23, 30, 37, 49] 成本：87.37

最优取点方案[4, 16, 26, 35, 48] 成本：84.75

最优取点方案[4, 20, 33, 47] 成本：100.58

由此可以看出来三次样条插值法能更好地降低成本；此外，虽然选取五个点可能不能做出较精细的拟合曲线，但计算成本最低，选取六、七个点虽说拟合曲线较为精准，但成本较高，而取四个点偏差又过大，所以综合来看，我们选择五个点的方案。

以上过程实现代码均附在附录中。

5 结论

(1) 根据程序运行结果来看，最优定标方案应为：测试点组合[4, 16, 26, 35, 48]，拟合方法为三次样条插值 spline，对于样本的定标成本为：84.75；

(2) 不同的拟合方法各有优劣之处，应当根据实际情况具体问题具体分析，选取合适的拟合方式进行曲线拟合；

(3) 通过对比，我们可以得出取点数目和成本不可以兼低，因为取点数目越多拟合曲线

越精确，但固有成本相对又提高了，所以必须兼顾来考虑，通过计算得出结论。

6 致谢

通过本课程的学习，我们小组对 MATLAB 的使用有了新的认识，在今后的学习生活中又掌握了一门新的有力的工具，因此感到十分高兴与感谢。同时提升了我们对线性拟合与最优思想的认识，最后感谢老师和助教在平时学习中对我们遇到的问题加以指导，使我们成功地完成了此次任务！

7 参考文献

[1] 上海交大电子工程系．统计推断在数模转换系统中的应用课程讲义[EB/OL].ftp://202.120.39.248.

[2] 百度百科模拟退火算法百科词条：http://baike.baidu.com/view/18185.htm

[3] http://www.cnblogs.com/heaad/archive/2010/12/20/1911614.html

[4] 百度文库《pchip 和 spline 的区别》

附录

一：模拟退火算法的实现

main.m

```
origin=dlmread('20150915dataform.csv');  %打开文件
X=origin(1:2:end,1:end);%将表格中的 x 分别取出
% Y=origin(2:2:end,1:end);%将表格中的 y 分别取出

s0=randperm(51);
sT=sort(s0(1:4));
num=0;
T=100; %初始温度
T_min=0.01; %末温度
r=0.97; %控制降温的快慢
cost_min=0;
cost_last=0;
cc=[];

tic;
while T>T_min

    num=num+1;
    sTmp=setdiff(s0,sT);
    Stmp=sTmp(:,randperm(47));
    S=sT;
```

```
    S(randperm(4,1))=sTmp(randperm(47,1));
    S=sort(S);


%%%%%%% 答案检验程序 2015-11-04 %%%%%%%%
my_answer=S;%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;
if num==1
       cost_min=cost;
       cost_last=cost;
       sT=S;
       S_min=S;
end
dE=cost_last-cost;
if dE>0;
    cost_min=cost;
    cost_last=cost;
    sT=S;
    S_min=S;
    else if exp(dE/T)>rand
           cost_last=cost;
           sT=S;
       end
end
    T=T*0.97;
   cc(num,:)=cost_min;
end
toc;
% 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_min);
S_min;
```

二：定标函数的实现

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
```

三：散点图绘制的实现

```
origin=dlmread('20150915dataform.csv');  %打开文件
x=origin(1:2:2,1:end)

y=zeros(400,51);              %创建备用矩阵
ranX=zeros(1,6);
ranY=zeros(1,6);


for i=1:400      % 读入 y
    y(i,:)=origin(2*i,:);
end

R1=randint(1,4,[1,400])     %随机取四组点绘制散点图

for i=1:1:4
    Y_1=y(R1(i),:);
    figure(i);
    scatter(x,Y_1,'.','k');
end
```