统计推断在数模系统中的应用

组号：25　　组长：彭长欢　514030913　　组员：王佳槐　5140309141

摘要：通过遗传算法和多种拟合方法对大量数据进行分析，为一个内部监测某项与外部环境有关的物理量进行定标，寻找最优化的校准工序方案。遗传算法是一种启发式算法，能够得到最优的近似解。本文以遗传算法为基础，采用三次样条插值法进行拟合，得出最优方案。

关键词：遗传算法，三次样条插值法，拟合

ABSTRACT: The author analysed the Genetic Algorithm as well as many fitting methods in order to search for a calibration for an interior monitoring physical quantity which is related to the outside conditions and seek a method for the optimized calibration procedure.The GA can be regard as a heuristic algorithm which can get the optimized approximate solution.In order to get optimized method the article based on the GA use cubic spline interpolation.

KEY WORDS:the Genetic Algorithm,cubic spline interpolation,fitting

1 引言

随着电子科技的发展，电子产品已经渗透进人们的生活，而对于一个好的电子产品来说，校准和定标的重要性是不言而喻的。校准和定标的工作一般是先生产一批量的电子产品，测得若干组数据，再进行一定的处理获得函数关系，以达到最终定标的目的，但在实际的生产应用中，我们很难找到一个对应的数学函数关系式去描述一个电子产品的输入端和输出端之间的关系，因为输入数据与输出数据之间一般呈非线性关系，即对应的图像大多均为曲线，因此在进行数据处理时需要进行一定的拟合，而对于数据如何选取和监测对于生产商来说是需要付出一定的成本的，而如何在保证准确性的同时尽量少的测量数据即减小生产成本，可以通过算法设计出一个最优传感特性校准（定标工序）方案。

2 具体问题

2.1 研究对象

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。

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，获得信号作为 Y 的读数（监测模块对 Y 的估测值）。

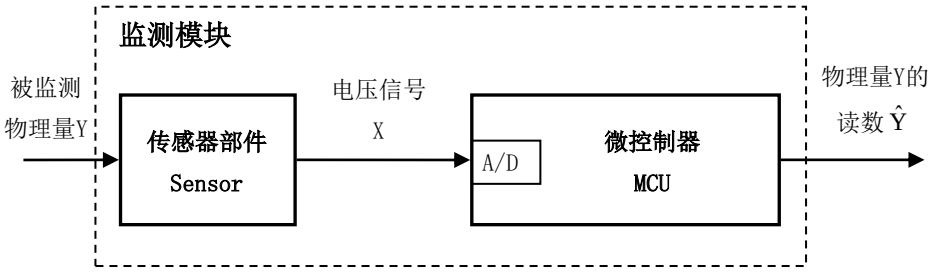

图 1 监测模块组成框图

我们需要建立一个数学模型，对一件物理量进行限定，这样才能进行有效讨论。

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 课题研究的主要步骤

（1）初步分析样本数据

（2）比较不同的拟合方法

（3）使用遗传算法，用三次样条插值法找到一个定标准确度足够高的方案

（4）拓展不同的算法

3 样本数据分析

原始数据一共有 400 个样本，从这些样本中随机抽取了 10 个样本，根据 X,Y 的取值在 Origin 中作出了散点图像。如图 2 所示，从图像上我们大致可以得知：

（1）Y 的取值随着 X 呈单调递增的趋势

（2）不同的样本的特性曲线形态相似但两两相异

（3）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）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；

（5 ）不同个体的中段起点位置、终点位置有随机性差异。

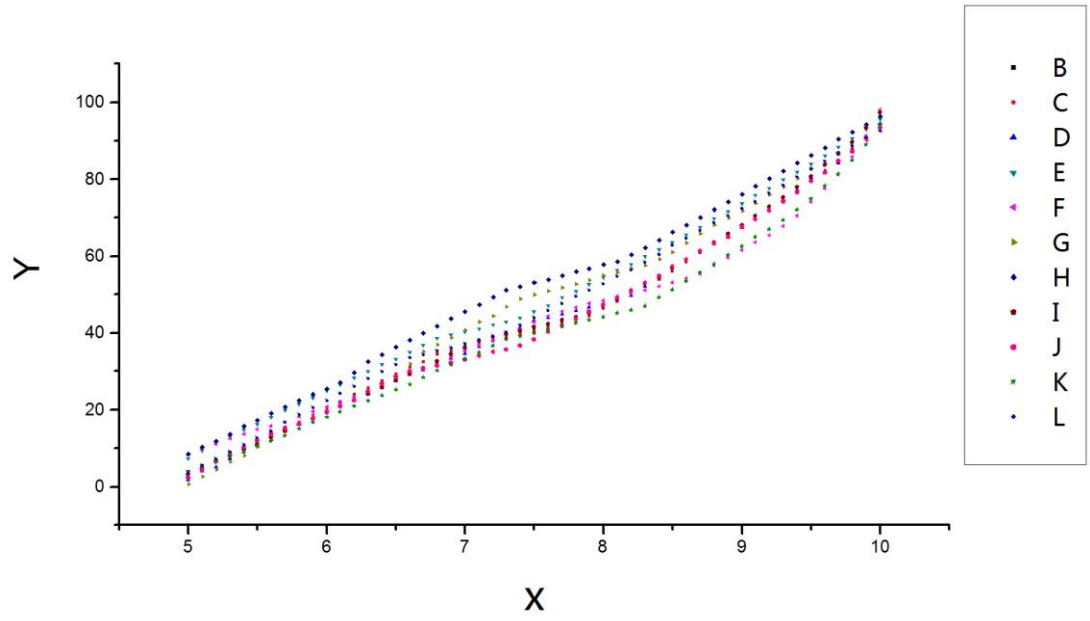

图 2　10 个样本的 X-Y 特性图

4 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

$$s_{i,j}=\begin{cases}0 & \text{if } \left|\hat{y}_{i,j}-y_{i,j}\right|\le 0.4\\ 0.1 & \text{if } 0.4<\left|\hat{y}_{i,j}-y_{i,j}\right|\le 0.6\\ 0.7 & \text{if } 0.6<\left|\hat{y}_{i,j}-y_{i,j}\right|\le 0.8\\ 0.9 & \text{if } 0.8<\left|\hat{y}_{i,j}-y_{i,j}\right|\le 1\\ 1.5 & \text{if } 1<\left|\hat{y}_{i,j}-y_{i,j}\right|\le 2\\ 6 & \text{if } 2<\left|\hat{y}_{i,j}-y_{i,j}\right|\le 3\\ 12 & \text{if } 3<\left|\hat{y}_{i,j}-y_{i,j}\right|\le 5\\ 25 & \text{if } \left|\hat{y}_{i,j}-y_{i,j}\right|>5\end{cases} \qquad (4\text{-}1)$$

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。

- 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

$$S_i=\sum_{j=1}^{51}s_{i,j}+q\cdot n_i \qquad (4\text{-}2)$$

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。

- 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。

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

总成本较低的校准方案，认定为较优方案。

5 拟合方法选取与讨论

拟合，就是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。

5.1 三次多项式拟合

1、由图 2 可知，曲线可分为首、中、尾三段，中段的平均斜率明显低于首尾两段，符合三次多项式的图像特征，所以我们首先想到了使用三次多项式的方法进行拟合。三次多项式选取七个特征点即可进行拟合。样本为 S={S1,S2.....S6,S7}。

$$Y=a_3X^3+a_2X^2+a_1X+a_0 \qquad (5\text{-}1)$$

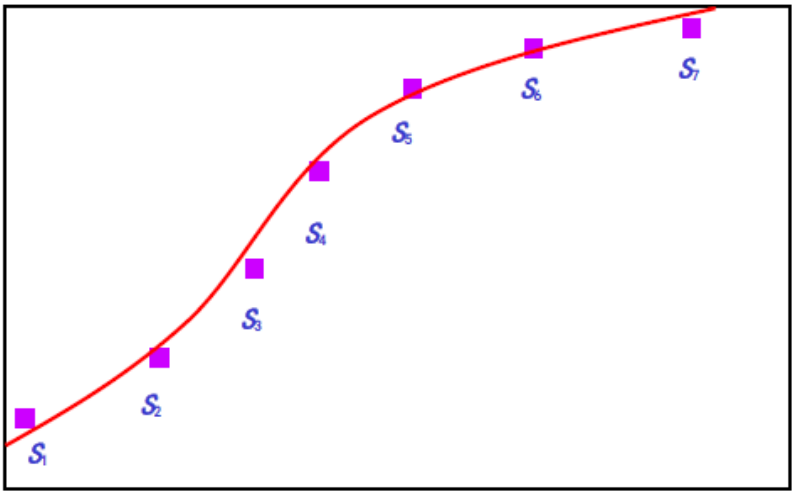

图 3　三次多项式拟合示意图

2. 曲线的拟合度。将去掉 1 和 51 号点的其余 49 个点的输出值代入上述三次多项式计算出相应个输出值，通过评价函数得到评价分值。

3. 实现最优化过程，即不断的寻找存在的可能解，继续前两步骤。每次得到可能解，通过评价，做出一定概率的接受或者舍弃，实现当前解的优化，直至达到终止条件。

5.2 三次样条插值拟合

1、插值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。

2、使用三次样条插值对确定一个可能解，即 7 个特征点组合 S={s1, s2, …, s7}进行拟合。具体方法为：首先对于非两端点，以四个连续点确定一条三次曲线，但仅在中间两点之间用

该三次曲线表示，以此类推，所有非两端点之间均有三次曲线。两端点由端点处三个点用二次曲线拟合。

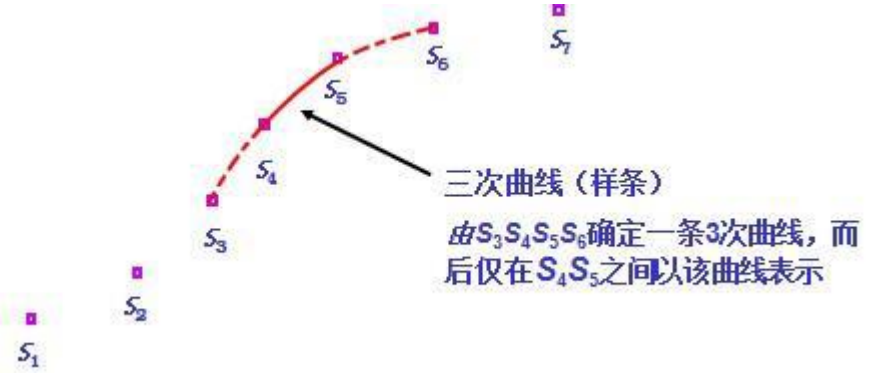

图 4　三次样条插值拟合示意图

3、评估曲线的拟合度。去掉首尾两点对中间 49 个点进行定标，计算出总体成本，通过评价函数得出拟合效果的好坏程度。

4、通过不断的寻找可能解，重复第 2、3 步骤。得到每次的可能解，通过评价，做出一定的概率的接受和舍弃，实现对当前解的优化，最终达到终止条件。

6 算法的选取与讨论

6.1 穷举法

穷举法的基本思想是根据题目的部分条件确定答案的大致范围，并在此范围内对所有可能的情况逐一验证，直到全部情况验证完毕。若某个情况验证符合题目的全部条件，则为本问题的一个解；若全部情况验证后都不符合题目的全部条件，则本题无解。

而基于以上拟合方法所讨论的 7 个点，对于穷举算法一共需要 $C_{51}^{7}=1,15775100$ 次，这个数字对于当前的计算机的计算速度来说，并不是不可解问题，但这会耗费大量的资源与优化该问题的初衷相违背，所以穷举法对于我们来说并不具有参考意义。

6.2 遗传算法

遗传算法（Genetic Algorithm）是模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型，是一种通过模拟自然进化过程搜索最优解的方法。相对来说代码简短，计算时间短，结果偶然性较小。因此我们在本课程中采取遗传算法。

遗传算法的基本运算过程如下：

a)初始化：设置进化代数计数器 t=0，设置最大进化代数 T，随机生成 M 个个体作为初始群体 P(0)。

b)个体评价：计算群体 P(t)中各个个体的适应度。

c)选择运算:将选择算子作用于群体。选择的目的是把优化的个体直接遗传到下一代或通过配对交叉产生新的个体再遗传到下一代。选择操作是建立在群体中个体的适应度评估基础上的。

d)交叉运算：将交叉算子作用于群体。遗传算法中起核心作用的就是交叉算子。

e)变异运算：将变异算子作用于群体。即是对群体中的个体串的某些基因座上的基因值作变动。

群体 P(t)经过选择、交叉、变异运算之后得到下一代群体 P(t+1)。

f)终止条件判断:若 t=T,则以进化过程中所得到的具有最大适应度个体作为最优解输出，终止计算。

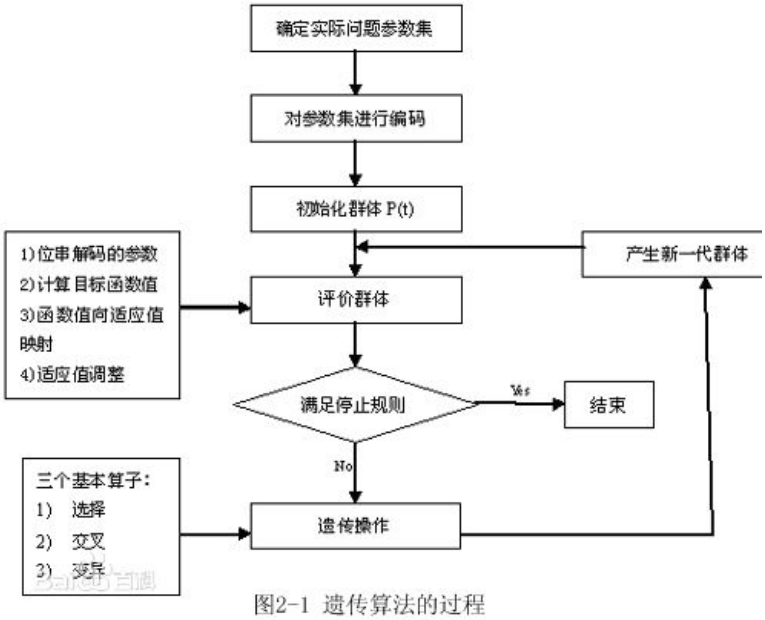

图2-1 遗传算法的过程

7 程序设计思想

7.1 主程序

（1）设置初值：种群最大数量 pop_size=100,终止进化代数 generation_size=100，交叉概率 cross_rate=0.88,mutation_rate=0.01。再将读入的数据存入一个 400×51 的 zeros 矩阵

（2）随机生成：geneinit(100)函数用于随机生成每个个体的基因组成（第 1 和 51 个基因不选），用 round(rand(100,51)-0.2);对每一行生成比特数，1 表示选择该数据点，0 则不选。对此进行 100 代遗传循环。

（3）遗传循环体：find(gene(1,:)==1)函数用于观察输出的每代最优个体的基因组成（也即所选取点的编号），adapt()用于个体适应度的生成（每个选点方案的平均成本）。然后进行选择 select()、交叉 generate()、变异 mutate()。

7.2 个体适应度

个体适应度由 geneinti（）函数进行初始化。生成一个 100×1 的矩阵存储 100 个不同选点方案的平均成本。

7.3 选择运算以及生存概率的确定

得到每个个体的平均成本后先用最大平均成本个体与每个个体之差求和得到基数，再用每个个体与最大个体的差值除以基数，得到一个相加得 1.0 的个体种群生存概率。

其后用 sortrows()对每个个体按生存概率从小到大排列。

7.4 交叉的实现

选择出新种群父代后，在父代中进行交叉。本例采用单点交叉即在基因段中从随机一点断裂，并与另一基因进行重组的方法。交叉概率定为 0.88，通过生成 0 到 1 的随机数，若小于 0.88 则进行交叉来实现。将父代随机排列，并进行编号，按照 i 和（pop_size-i+2）的规则结对。若判定二者进行交叉，则随机产生一个基因断裂点，将二者基因重组后输入新的种群。若判定二者不进行交叉，则直接将二者输入新的种群。

7.5 变异的实现

变异就是指将基因中 0、1 进行互换。首先给定基因变异的概率，因为变异概率只是提供多样化的选择，不能因为变异而改变基因整体的优化性，因此变异的概率应该较小，定为 0.01。而种群中每一个个体的每一个基因均有可能发生突变。鉴于种群是用二维矩阵表示，

所以变异的列举应该相应地使用二重循环进行实现。

8 运算结果分析

8.1 程序运行结果图示（100 次）

```
point: [  2 10 20 26 33 43 50 ]    cost: 95.229000
point: [  3 10 22 29 34 43 50 ]    cost: 97.398250
point: [  3  9 21 29 41 49 ]    cost: 96.482500
point: [  4 13 22 31 41 50 ]    cost: 97.676750
point: [  2  9 19 26 32 43 50 ]    cost: 95.286000
point: [  4  9 18 28 36 45 49 ]    cost: 99.066500
point: [  4 11 21 30 41 49 ]    cost: 96.435250
point: [  4  8 22 25 33 41 50 ]    cost: 99.850750
point: [  2 10 21 29 36 46 50 ]    cost: 97.447500
point: [  2  9 21 27 34 44 50 ]    cost: 95.522250
```

成本约为41.89802时，我们得出了校准方案，在[3 12 22 31 43 50] 定标更好，最小

9 拓展算法

用遗传算法得到了最小成本后，我们发现与遗传算法相同的这类启发式算法还有模拟退火算法。

模拟退火算法是根据 Metropolis 准则，粒子在温度 T 时趋于平衡的概率为 e(-ΔE/(kT))，其中 E 为温度 T 时的内能，ΔE 为其改变量，k 为 Boltzmann 常数。在这个算法中，用内能 E 来模拟目标函数，温度 T 为控制参数。从初始解开始，对当前解不断产生新解，计算目标函数的插值，接受或舍弃新解的迭代步骤。t 的值不断减小，当温度降到某一特定值后，算法结束，停止迭代，得到近似的最优解。

遗传算法主要由选择，交叉，变异等操作组成，而模拟退火算法是采用单个个体进行进化，不断的通过迭代得到最优解，只有当新解优于当前解时才接受新解。这样所花费的时间也较长，而遗传算法则是通过选择操作选得不同的个体，进行交叉变异产生新个体，从而种群不断进化进行最优解的选择，遗传算法只能得到一个广阔的近似解空间，解的选择性较多，但时间较短是遗传算法的一大优势。

10 参考文献

[1]袁炎：统计推断讲座

[2]http://baike.baidu.com

[3]2013 级第 7 组和第 43 组统计推断报告

[4]https://pt.sjtu.edu.cn/login.php（葡萄网） [视频][MATLAB 数值分析与应用（第 2 版）]（matlab 教学视频）

附录：

程序代码：

（1）主程序

```
data=csvread('20150915dataform.csv');
pop_size=100;       %种群最大数量
cross_rate=0.88;    %交叉概率
mutation_rate=0.01;   %突变概率
generation_size=100;      %进化代数
y=zeros(400,51);
y(1:400,:)=data(2:2:800,:);
gene=geneinit(pop_size);         %初始化每一代的个体数
for g=1:generation_size
    display(g);
    cost=adapt(gene,y,pop_size);    %个体数拟合
    gene=select(gene,cost,pop_size);     %选择存活的个体
    gene=generate(gene,pop_size,cross_rate);  %产生下一代个体
    gene=mutate(gene,pop_size,mutation_rate);  %对变异的个体进行处理
    display(find(gene(1,:)==1));
end
xx=find(gene(1,:)==1);
assess(xx,y);
```

（2）

```
function out = geneinit(pop_size)   %随机产生初始的种群

out=round(rand(pop_size,51)-0.2);

out(:,1)=0;

out(:,51)=0;


end
```

（3）

```
function out = adapt(gene,y,pop_size) %计算每个个体的平均成本
out=zeros(pop_size,1);
x=5:0.1:10;
for i=1:pop_size
    c=sum(gene(i,:)==1);       %测试点数量
    pos=find(gene(i,:)==1);    %测试点位置
    xx=5+(pos-1)*0.1;          %测试点 X 值
    yy=y(:,pos);               %测试点 Y 值
    f=spline(xx,yy);
    difference=ppval(f,x)-y;
    out(i)=12*c+errorcost(difference)/400;
end
min(out)
mean(out)
```

```
end
```

（4）

```
function out = select(gene,cost,pop_size)   %选择

out=zeros(pop_size,51);
cost0=max(cost)-cost;
s0=sum(cost0);
cost0=cost0/s0;
s=[[1:pop_size+2]',zeros(pop_size+2,1),zeros(pop_size+2,1)];
s(:,1)=s(:,1)-1;
s(pop_size+2,3)=1;
s(2:pop_size+1,3)=cost0;
s=sortrows(s,3);
s(pop_size+2,2)=1;
for i=2:pop_size+1
    t=rand();
    j=search(t,s,1,pop_size+2);
    out(i,:)=gene(j,:);

end
sort0=[[1:pop_size]',cost];
sort0=sortrows(sort0,2);
out(1,:)=gene(sort0(1,1),:);


end
```

（5）

```
function out = generate(gene,pop_size,cross_rate)   %交叉

for i=2:floor(pop_size/2+1)        %除第一行外的所有个体进行交叉
    out=gene;
    mid=floor(rand()*50)+1;
    t=rand();
    if t<=cross_rate
        out(i,1:mid)=gene(pop_size-i+2,1:mid);
        out(pop_size-i+2,1:mid)=gene(i,1:mid);
        out(i,mid+1:51)=gene(pop_size-i+2,mid+1:51);
        out(pop_size-i+2,mid+1:51)=gene(i,mid+1:51);
    end
end
end
```

（6）

```
function out = mutate(gene,pop_size,mutation_rate)   %变异
```

```
out=gene;
for i=2:pop_size;      %第一个个体不参加变异
    for j=2:50;
        t=rand();
        if t<=mutation_rate
            out(i,j)=~out(i,j);
        end
    end
end
end
```

（7）
```
function [out] = search(in,s,l,r)    %使用二分法进行查找

mid=floor((l+r)/2);
if in<=s(mid)
    if in>s(mid-1)
        out=mid-1;
    else
        out=search(in,s,l,mid);
    end
else
    if in<=s(mid+1)
        out=mid;
    else
        out=search(in,s,mid,r);
    end
```

（8）
```
function [out] = errorcost(difference)    %误差成本计算函数

t=abs(difference);
t0=sum(sum(t<=0.4));
t1=sum(sum(t<=0.6))-t0;
t2=sum(sum(t<=0.8))-t0-t1;
t3=sum(sum(t<=1))-t0-t1-t2;
t4=sum(sum(t<=2))-t0-t1-t2-t3;
t5=sum(sum(t<=3))-t0-t1-t2-t3-t4;
t6=sum(sum(t<=5))-t0-t1-t2-t3-t4-t5;
t7=sum(sum(t>5));

out=0.1*t1+0.7*t2+0.9*t3+1.5*t4+6*t5+12*t6+25*t7;

end
```

（9）
```
function [out] =assess(in,y)      %计算成本
out=length(in)*12;
```

```
x=5:0.1:10;
xx=5+(in-1)*0.1;
yy=y(:,in);
f=spline(xx,yy);
difference=ppval(f,x)-y;
out=out+errorcost(difference)/400;

s=sum(difference.^2);
fid=fopen('answer.txt','a');          %格式化输出
fprintf(fid,'\n point: [ ');
fprintf(fid,'%2d ',in);
fprintf(fid,']    cost: %7f\n\n',out);
fclose(fid);
end
```

检验程序

```
my_answer=[  3 ,12 ,22, 31 ,43 ,50 ];
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
       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);


function y1 = mycurvefitting( x_premea,y0_premea )

x=[5.0:0.1:10.0];

y1=interp1(x_premea,y0_premea,x,'spline');

end
```