统计推断在数模转换系统中的应用

组号 66　　姓名：陈丹旸　　学号：5130309354

摘要：本文是上海交通大学电子信息与电气工程学院课程设计《统计推断在数模转换系统中的应用》的课程论文。对于某种检测模块，其输入与输出存在着一定关系，这种关系呈明显的非线性且个体间差异较大。但在实际过程中，如果对每个样品均密集选点测定，则将消耗大量的时间和经济。因此我们选择减少测定次数以提高效率。在选取 4 到 10 个测定点的情况下分析各种估测方法所需成本，并选出成本最低的方案。

关键词：统计推断，选点估测，启发式搜索，多项式拟合

Application of Statistical Inference in DA Inverting System

ABSTRACT: This article is for Course Design-Application of Statistical Inference in DA Inverting System, School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University. There exists certain relationship between input and output in a sensor device, which is nonlinear and has apparent individual difference. However, in engineering practice, it will spare lots of time and money to choose numerous sample to measure. In this way, reducing sample points to improve efficiency is what we need. In this article, we choose 4 to 10 sample points to calculating cost of different estimating methods and put forward the method which costs the least.

Key words: Statistical Inference, Sample estimation, Heuristic method, Cubic spline interpolation, the polynomial fitting

1. 引言

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（传感器部件监测的对象物理量）与值（X 传感部件的输出电压信号）间一一对应的特性关系的过程。在工程实践与科学实验中往往是通过多次测试得到一系列的数据，这些数据很难找到非常精确描述两个变量的函数。通常采取的方法是通过多种不同曲线拟合，算出多种不同的方案，并选择某一统计量作为评定优劣标准，从而选出最优的方案。

2. 问题的提出

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。前期已经通过试验性小批量生产，制造了一批传感部件样品，并通过实验测定了每个样品的特性数值。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。

2.1 数据来源

监测模块的组成框图如图 1。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）。

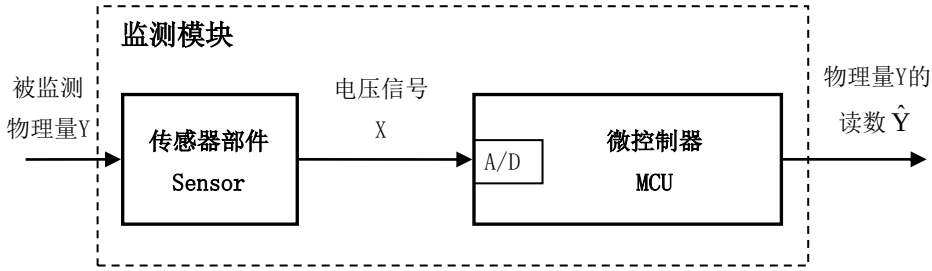

图 1 监测模块组成框图

2.2 研究装置及已得数据介绍

这种电子装置有 51 个有效控制点，对应输出电压 U= 5.0V 5.1V …10.0V。实际制作了 469 个此种装置样品，并通过实验测量，获得了每个样品的特性数据。

2.3 数学模型

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

$$s_{i,j}=\begin{cases}0 & \text{if } \left|\hat{y}_{i,j}-y_{i,j}\right|\le 1\\ 1 & \text{if } 1<\left|\hat{y}_{i,j}-y_{i,j}\right|\le 2\\ 2 & \text{if } 2<\left|\hat{y}_{i,j}-y_{i,j}\right|\le 3\\ 4 & \text{if } 3<\left|\hat{y}_{i,j}-y_{i,j}\right|\le 5\\ 10 & \text{if } \left|\hat{y}_{i,j}-y_{i,j}\right|>5\end{cases} \qquad (1)$$

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=20。

- 某一样本个体的定标成本

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

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。

- 校准方案总体成本

按式（3）计算评估校准方案的总体成本，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。

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

总体成本较低的校准方案，认定为较优方案。

在遗传算法中，由于需要取成本较低的方案但该算法为择适应度高者存活，因此适应度

函数被设为成本的倒数。但是在模拟退火算法中，则将温度函数直接设置为成本函数。根据测试显示，当循环次数为 50 次时遗传算法所得结果可基本接近最优解，当循环次数为 100 次时，模拟退火算法中的温度可基本不变，也即最优解。

为减少计算量，我随机取出了 16 个样本进行计算，使用 randperm 函数打乱顺序并取其前 16 个样本。

3. 暴力穷举法

获得全局最优解的方法是将所有可能性进行计算，以取七个点为例，总共需要尝试（51 中取 7 的组合）共有 115775100 次，计算工作量十分巨大，通过观察发现特征点之间呈离散分布，所以在全局范围内选点会有大量的不必要计算。为了减少计算量，可以对每个点划分范围从而优化算法。通过缩小范围的方法找到拟合点的大致位置，选取处于端点的第 1 和第 51 个数据，并且规定任意两点之间的间距不能小于 5，然后使用随机选取的方法获得取点位置，计算其总体成本，并选取最低的方案。

选取区间[1,51]再利用循环来找到使平均成本最低的观测点（程序及运行结果见附件）。

这种方案在所有方法中是最简单的，但其时间复杂度过高，以至于其较高的精度无法弥补这一缺陷。

4. 遗传算法

4. 1 遗传算法概述

遗传算法是一类借鉴生物界的进化规律（适者生存，优胜劣汰遗传机制）演化而来的随机化搜索方法。

遗传算法是模拟达尔文的遗传选择和自然淘汰的生物进化过程的计算模型。它的思想源于生物遗传学和适者生存的自然规律，是具有“生存＋检测”的迭代过程的搜索算法。遗传算法以一种群体中的所有个体为对象，并利用随机化技术指导对一个被编码的参数空间进行高效搜索。其中，选择、交叉和变异构成了遗传算法的遗传操作；参数编码、初始群体的设定、适应度函数的设计、遗传操作设计、控制参数设定五个要素组成了遗传算法的核心内容。

根据进化术语，对群体执行的操作有三种，遗传算法流程如图 2。

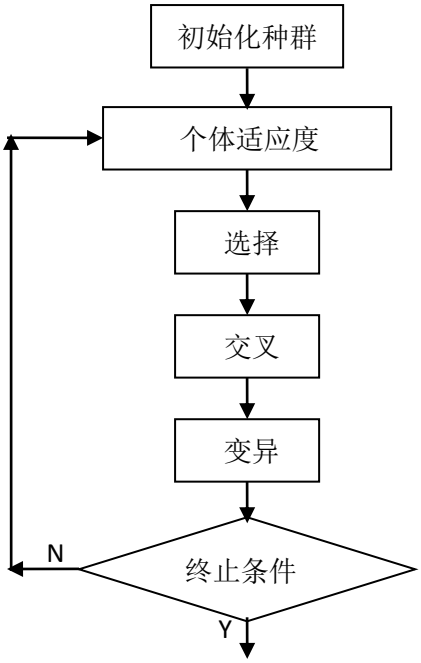

图 2 遗传算法流程图

4. 2 遗传算法的具体思路

4.2.1 参数设置

初始种群个体：16 个

遗传代数：50

交叉位置：第四位

变异概率：0.1%

首先创建拥有 16 个随机样本的种群，并计算其适应度，接着随机变异，变异率为 0.1%，然后进行交叉并算出新个体的适应度，与原个体相比较取其高者留下，并重复 50 次，在最终所得到的种群中取所有样本计算成本。

在曲线模拟上我使用了 polyfit 函数拟合 3 次函数，得到 3 次函数的各个参数，带入适应度函数中算出各点的适应度，之后在第四点进行交叉，形成新种群，继续计算适应度并留下适应度较高的解形成新种群，并循环 50 次，最后留下的种群计算成本并显示。

4.2.2 运行结果

利用遗传算法，得到的最低误差组为[1 6 11 28 32 47 51]，经计算得出的最低成本为 1019.2。

5. 模拟退火算法

5. 1 模拟退火算法概述

模拟退火算法来源于固体退火原理，将固体加温至充分高，再让其徐徐冷却，加温时，固体内部粒子随温升变为无序状，内能增大，而徐徐冷却时粒子渐趋有序，在每个温度都达到平衡态，最后在常温时达到基态，内能减为最小。模拟退火算法流程如图 3，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，……，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。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。

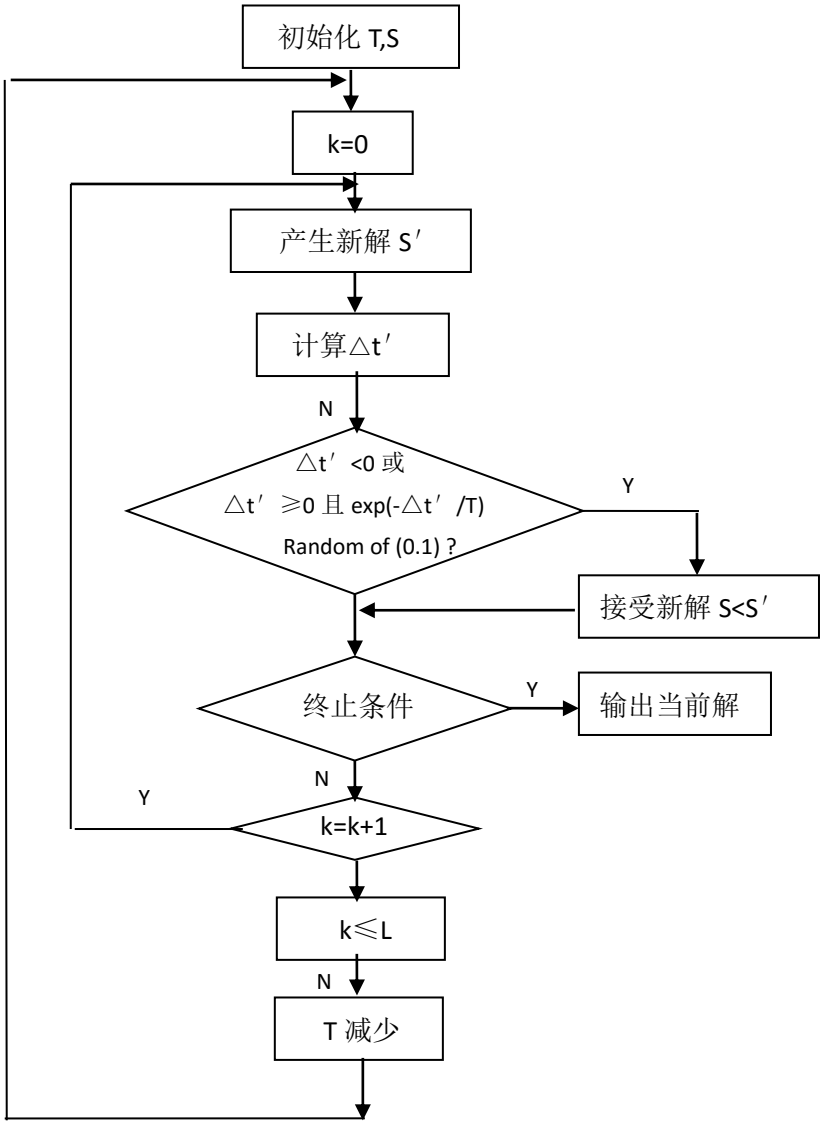

图 3 模拟退火算法流程图

5. 2 模拟退火算法的具体思路

5.2.1 参数设置

初始解状态：16 个随机样本

取点个数：7

温度函数：即总体成本

退火时间（即循环次数）：100 次

首先取 16 个随机样本作为初始解空间，然后随机取 7 个点作为最开始状态，并计算其温度，接着产生新解，计算其是否降温，若降温则将该解作为解空间的下一状态，若不降温则有 exp(-Δt/T)的可能性将该解作为解空间的下一状态。如此循环 100 次，温度将趋于平衡，该状态即为最佳解。最后显示出解空间及其成本。

5.2.2 运行结果

利用退火算法，得到最低误差组为[2　9　17　23　35　43　51],经计算得出最低成本为 1043.9。

6. 结论

1.暴力穷举法在计算机运算时太长，通过分析，并进行了初步尝试之后，只能列出代码，并没有得到最终结果。

2.利用遗传算法，得到的最低误差组为[1　6　12　29　32　47　51]，经计算得出的最低成本为 1028.3。

3.利用退火算法，得到的最低误差组为[2　9　17　23　35　43　51]，经计算得出最低成本为 1043.9。

4.在编程过程中，我认为遗传算法相比较于模拟退火算法而言更适合这个问题，模拟退火算法相对于遗传算法来说不确定性更大，也就是说，在产生新解的过程中，退火算法相对不可控，会经常出现背离最优解的新解，这在一定程度上拖长了获得最优解的时间，在实际情况下也证明了这一点。

6. 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] 统计推断课程 ppt

[3] 卓金武. MATLAB 在数学建模中的应用[M] .北京.北京航空航天大学出版社.2011

[4] 占君、张倩、满谦.MATLAB 函数查询手册[M].北京.机械工业出版社.2011

附件：代码

1.暴力穷举法

```
warning off all;
clear;
clc;
tic;
testdata = xlsread('data.xls','sheet 1','A1:AY938','basic');
min = 1026

for point = 1:4
    for p1 = 1:(52-point)
        for p2 = p1:(53-point)
            for p3 = p2:(54-point)
                for p4 = p3:(55-point)
                    if point>4
                    {
                        for p5 = p4:(56-point)
                            if point>5
                            {
                                for p6 = p5:(57-point)
                                    if point>6
                                    {
                                        for p7 = p6:(58-point)
                                            if point>7
                                            {
                                                for p8 = p7:(59-point)
                                                    if point>8
                                                    {
                                                        for p9 = p8:(60-point)
                                                            if point>9
                                                            {
                                                                for p10 =
p9:(61-point)
                                                                    testpoint =
[p1 p2 p3 p4 p5 p6 p7 p8 p9 p10]
                                                                    score =
curvefit(testdata,testpoint)
                                                                    if score<min
                                                                        min =
score

disp(testpoint)
                                                                    disp(min)
                                                                    end;
```

```
                                        }
                                        else
                                            testpoint = [p1
p2 p3 p4 p5 p6 p7 p8 p9]
                                            score =
curvefit(testdata,testpoint)
                                            if score<min
                                                min = score
                                            disp(testpoint)
                                            disp(min)
                                            end;
                                    }
                                    else
                                        testpoint = [p1 p2 p3 p4
p5 p6 p7 p8]
                                        score =
curvefit(testdata,testpoint)
                                        if score<min
                                            min = score
                                        disp(testpoint)
                                        disp(min)
                                        end;
                                }
                                else
                                    testpoint = [p1 p2 p3 p4 p5 p6
p7]
                                    score =
curvefit(testdata,testpoint)
                                    if score<min
                                        min = score
                                    disp(testpoint)
                                    disp(min)
                                    end;
                            }
                            else
                                testpoint = [p1 p2 p3 p4 p5 p6]
                                score = curvefit(testdata,testpoint)
                                if score<min
                                    min = score
                                disp(testpoint)
                                disp(min)
                                end;
                        }
                        else
```

```
                                testpoint = [p1 p2 p3 p4 p5]
                                score = curvefit(testdata,testpoint)
                                if score<min
                                    min = score
                                disp(testpoint)
                                disp(min)
                                end;
                            }
                            else
                                testpoint = [p1 p2 p3 p4]
                                score = curvefit(testdata,testpoint)
                                if score<min
                                    min = score
                                disp(testpoint)
                                disp(min)
                        end;
                    end;
                end;
            end;
        end;
toc;
```

2.遗传算法

主程序：

```
ran_samplenum=16;
num=7;
mess_sample=randperm(469);
ran_samsit=mess_sample(1,1:ran_samplenum);
[adapt]=Adapt(ran_point,num,ran_samsit(1,i),ran_samplenum,a);
mut=randi(1:1000);
mut_sam=randi(16);
if mut==1
    [ran_num(i,:)]=Mutation(ran_num(i,:));

alldata=xlsread('data.xls','20141010dataform','A1:AY938','basic');
    ran_point(line,1:num)=alldata(line1,ran_num(1,1:num));
end
for i=1:50
    cross_sample=randperm(16);
    for r=1:8

[ran_point1(cross_sample(2*r-1,:)),ran_point1(cross_sample(2*r,:))]=C
ross(ran_point(cross_sample(2*r-1,:)),ran_point(cross_sample(2*r,:)))
;
```

```
[ran_num1(cross_sample(2*r-1,:)),ran_num1(cross_sample(2*r,:))]=Cross
(ran_num(cross_sample(2*r-1,:)),ran_num(cross_sample(2*r,:)));

[adapt1]=Adapt(ran_point1,num,ran_samsit(1,i),ran_samplenum,b);
       [rest,IX]=sort([adapt(1,2*r-1) adapt(1,2*r) adapt1(1,2*r-1)
adapt(1,2*r)]);
       if rest(1,1)==adapt1(1,2*r-1)

ran_point(cross_sample(2*r-1,:))=ran_point1(cross_sample(2*r-1,:));

ran_num(cross_sample(2*r-1,:))=ran_num1(cross_sample(2*r-1,:));
          adapt(1,2*r-1)=adapt1(1,2*r-1);
       end
       if rest(1,1)==adapt1(1,2*r)

ran_point(cross_sample(2*r-1,:))=ran_point1(cross_sample(2*r,:));

ran_num(cross_sample(2*r-1,:))=ran_num1(cross_sample(2*r,:));
          adapt(1,2*r-1)=adapt1(1,2*r);
       end
       if rest(1,2)==adapt1(1,2*r-1)

ran_point(cross_sample(2*r,:))=ran_point1(cross_sample(2*r-1,:));

ran_num(cross_sample(2*r,:))=ran_num1(cross_sample(2*r-1,:));
          adapt(1,2*r)=adapt1(1,2*r-1);
       end
       if rest(1,2)==adapt1(1,2*r)

ran_point(cross_sample(2*r,:))=ran_point1(cross_sample(2*r,:));
          ran_num(cross_sample(2*r,:))=ran_num1(cross_sample(2*r,:));
          adapt(1,2*r)=adapt1(1,2*r);
       end
       disp(rest(1,1:2));
   end
end
[real,Ix]=sort(adapt(1,1:16));
situation=find(real(1,1));
disp(ran_num(situation,:));
```

随机点：

```
function[ran_point,ran_num]=Ranpoint(num,line)
mess=randperm(51);
ran_num=mess(line,1:num);
```

```
alldata=xlsread('data.xls','20141010dataform','A1:AY938','basic');
line1=line*2;
for i=1:num;
    ran_point(line,i)=alldata(line1,ran_num(line,i));
disp(ran_num);
disp(ran_point);
end;
end
```

适应度：

```
function[adapt]=Fitness(ran_point,num);
volt=xlsread('data.xls','20141010dataform','A1','basic');
parameter=polyfit(volt,ran_point,3);
S=0;
for i=1:num

cal_point=parameter(1)*(volt(i)^3)+parameter(2)*(volt(i)^2)+parameter
(3)*(volt(i))+parameter(4);
    if abs(cal_point-ran_point)<=1
        S=S;
    else if (abs(cal_point-ran_point)>1)&&(abs(cal_point-ran_point)<=2)
            S=S+1;
        else if
(abs(cal_point-ran_point)>2)&&(abs(cal_point-ran_point)<=3)
                S=S+2;
            else if
(abs(cal_point-ran_point)>3)&&(abs(cal_point-ran_point)<=5)
                    S=S+4;
                else if (abs(cal_point-ran_point)>5)
                        S=S+10;
                    end;
                end;
            end;
        end;
    end;
end;
adapt=S+num*20;
end
```

适应度计算：

```
function[adapt]=Adapt(ran_point,num,ran_samsit,ran_samplenum,b)
for i=1:ran_samplenum
    line=2*ran_samsit(1,i);
    [ran_point(i,:),ran_num(i,:)]=Ranpoint(num,line);
    [adapt(1,i)]=Fitness(ran_point,num);
    adapt(2,1)=i;
```

```
    adapt(2,2)=b;
    disp(adapt);
end
```

交叉：

```
function[new_point1,new_point2]=Cross(point1,point2)
new_point1(1,1:3)=point1(1,1:3);
new_point2(1,1:3)=point2(1,1:3);
new_point1(1,4:)=point2(1,4:);
new_point2(1,4:)=point1(1,4:);
```

变异：

```
function [new_num]=Mutation(num)
len=length(num);
sit=randi([1,len]);
new_num(1,1:len)=num(1,1:len);
new_num(1,sit)=num(1,sit);
disp(new_num);
```

3.模拟退火算法

```
warning off all;
clc;
tic;
num=7;
samplenum=16;
cycle=50;
volt=xlsread('data.xls','20141010dataform','A1:AY938','basic');
orisample=randperm(469);
ransample(1,1:samplenum)=orisample(1,1:samplenum);
oripoint=randperm(51);
ranpoint(1,1:num)=oripoint(1,1:num);
for z=1:num
    ran_point(1,z)=4.9+0.1*ranpoint(1,z);
end;
for c=1:samplenum
    line=2*ransample(1,c)
    parameter=polyfit(volt(line,:),ran_point,3);
    S=0;
    for i=1:num
        cal_point=parameter(1)*(ran_point(i)^3)+parameter(2)*(ran_poi
nt(i)^2)+parameter(3)*(ran_point(i))+parameter(4);
        if abs(cal_point-volt(line,ranpoint(num)))<=1
            S=S;
        else if (abs(cal_point-volt(line,ranpoint(num)))>1)&&(abs(cal
_point-volt(line,ranpoint(num)))<=2)
                S=S+1;
            else if (abs(cal_point-volt(line,ranpoint(num)))>2)&&(ab
```

```
s(cal_point-volt(line,ranpoint(num)))<=3)
                        S=S+2;
                    else if (abs(cal_point-volt(line,ranpoint(num)))>3)
&&(abs(cal_point-volt(line,ranpoint(num)))<=5)
                            S=S+4;
                        else if (abs(cal_point-volt(line,ranpoint(num)
))>5)
                                S=S+10;
                            end;
                        end;
                    end;
                end;
            end;
        end;
        adapt(1,c)=S+num*20;
end;
for m=1:100
    oripoint1=randperm(51);
    ranpoint1(1,1:num)=oripoint1(1,1:num);
    for y=1:num
        ran_point1(1,y)=4.9+0.1*ranpoint1(1,y);
    end;
    for b=1:samplenum
        line=2*ransample(1,b)
        parameter=polyfit(volt(line,:),ran_point1,3);
        S=0;
        for a=1:num
            cal_point=parameter(1)*(ran_point1(a)^3)+parameter(2)*(ra
n_point1(a)^2)+parameter(3)*(ran_point1(a))+parameter(4);
            if abs(cal_point-volt(line,ranpoint1(num)))<=1
                S=S;
            else if (abs(cal_point-volt(line,ranpoint1(num)))>1)&&(ab
s(cal_point-volt(line,ranpoint1(num)))<=2)
                    S=S+1;
                else if (abs(cal_point-volt(line,ranpoint1(num)))>2)&
&(abs(cal_point-volt(line,ranpoint1(num)))<=3)
                        S=S+2;
                    else if (abs(cal_point-volt(line,ranpoint1(num)))
>3)&&(abs(cal_point-volt(line,ranpoint1(num)))<=5)
                            S=S+4;
                        else if (abs(cal_point-volt(line,ranpoint1(nu
m)))>5)
                                S=S+10;
                            end;
```

```
                        end;
                    end;
                end;
            end;
        end;
        adapt(2,b)=S+num*20;
    end;
    for d=1:samplenum
        if adapt(2,d)<adapt(1,d)
            ranpoint(1,1:num)=ranpoint1(1,1:num);
            adapt(1,d)=adapt(1,d);
        else
            per=e^((adapt(2,d)-adapt(1,d))/adapt(1,d));
        end;
    end;
end;
disp(ranpoint);
disp(adapt);
```