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73组,成员罗毅学号5140309211, 周圣宇学号5140309213,在报告编写过程中，以下方面参考了往届报告，现列表说明：

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》，统计推断-第02组（张超）课程设计报告 参考了其中的算法思想，对代码内容做出了修改
算法描述方面	《统计推断在数模转换系统中的应用》，统计推断-第20组（艾玥祺）课程设计报告 参考了该组报告的算法思想描述，对叙述做出了修改。

除了以上注明的参考内容，和报告中列出的引用文献注解，本报告其他部分都不包含任何其他个人或集体已经发表或撰写过的作品成果。

计推断在数模转换系统中的应用

组号 73　　姓名 罗毅 学号 5140309211，姓名 周圣宇 学号 5140309213

摘要：本次课题是研究某监测模块中传感器部件监测的对象的物理量与传输部件的输出电压信号之间的关系。本文则是通过对原始数据的研究，通过暴力算法和遗传算法对问题的解决进行了探究。

关键词：暴力算法，模拟遗传算法。

1 引言

本次课题研究的是某监测模块中传感器部件监测的对象的物理量 y 与传输部件的输出电压信号 x 之间的关系。由于元件等各方面因素，每个元件的 y 与 x 的对应关系都不相同，如果逐个的去分析各个监测模块的性质，必定会耗费大量的人力物力，故通过剔除一些点，选取某些具有代表性的点，从而确定出监测模块相应参数之间的关系，便引出了该课题研究。

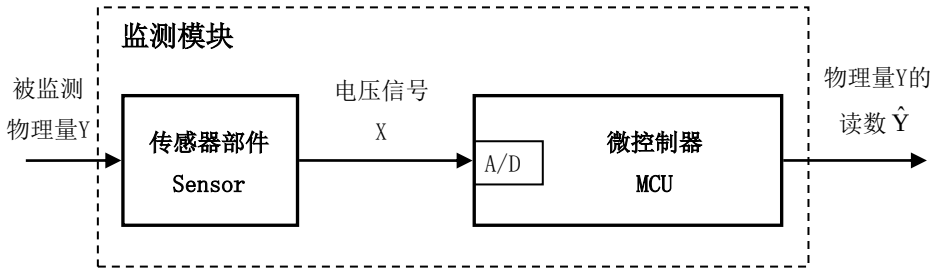


图 1 监测模块的组成框图

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 任务概述

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。

3 设计思路

基本任务中主要有两个方面的问题需要考虑：

一是如何选取 7 个特征点，虽然说暴力穷举法从理论上来说是可行的，但对于我们所使用的个人电脑来说运行所需要的时间过于长，并且从 51 个点中枚举所有 7 个特征点的组合来尝试拟合显然没有必要，因此使用适当的组合优化方法是必要的。二是用何种方法还原曲线的表达式，这里采用三次多项式拟合法以及三次样条插值法。

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

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

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。

5 拟合算法

1. 三次样条插值法

（1）方法介绍

插值法的基本思想是构造一个简单函数 y=G(x)作为 f(x)的近似表达式，以 G(x)的值作为函数 f(x)的近似值，并且要求 G(x)在给定数据点 x_i 的值与 f(x)的值相同，即 $G(x_i)=f(x_i)$，通常称 G(x)为 f(x)的插值函数，x_i 为插值节点。

（2）样本数据拟合

$$y=\text{spline}(x_0,y_0,x) \quad (5\text{-}1)$$

我们使用 MATLAB 中内嵌函数（式 5-1）对样本数据库中的某个数据点进行三次插值拟合，所得拟合曲线如图 5-1。

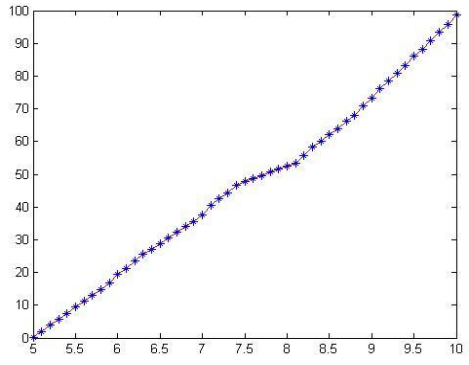


图 5-1 三次样条插值拟合曲线

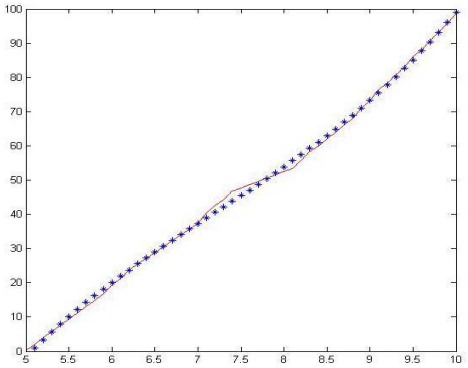


图 5-2 三次多项式拟合曲线

2. 三次多项式拟合法

（1） 方法介绍

多项式拟合是一种最基本的拟合方式，对于给定数据(x_i，y_i)，在给定的函数簇 Φ 中，求属于 Φ 的 f(x)，使误差 $r_i=f(x_i)-y_i$ 最小，这样所求的函数 f(x)即为拟合出来的多项式函数。使用三次多项式拟合，其 $\Phi(x)=ax^3+bx^2+cx+d$。

（2） 样本数据拟合

我们使用 MATLAB 中内嵌函数对样本数据库中的某个数据点进行三次多项式拟合，所得拟合曲线：

$$y=polyfit(x_0,y_0,m)$$

用 MA 其中 x_0, y_0 为拟合点坐标，m 为多项式拟合的次数，这里 m=3。

6 实验进程

1.暴力算法：

初步尝试暴力算法后发现运行时间过长，无法得到结果。所以暴力算法经测试后被我们排除在外。

2.遗传算法

（1）算法设计思想

概述;遗传算法(poptic Algorithm，GA 算法)是由 Holland 教授于 20 世纪 60 年代提出一种模拟达尔文生物进化论中的生物遗传进化过程和自然选择的计算模型．其原理是通过模拟生物进化的过程，使得在每一代中通过对种群进行杂交运算以及变异算子来产生新的个体，通过达尔文生物进化论中的自然选择原则来选择个体得到新的种群，使得每代中最好的个体的适应值越来越好，而个体的平均适应值也会越来越好。

算法设计过程：

根据我们所查到的资料，我们已经明白了利用遗传算法解决实际问题的核心思想：针对所要解决的问题，寻求一种利用数字编码来表示实际情景的方案。接下来我们需要得到一个初始的族群（利用随机数生成）用于“进化”。接着我们需要评估族群中的个体存活的概率（需要我们自己设计一个适应度函数）来模仿自然选择过程。接着还需要得到新个体：通过

交叉（模仿自然界的交配）以及变异（基因变异）。于是下一代族群由上一代个体的适应度（越高越有可能留下更多的后代）以及交叉和变异决定。反复进行这个过程，个体的平均适应值会越来越好。

基于这种考虑，我们设计遗传算法的各步骤如下：

初始化种群：

首先我们确定了数字化编码问题的方案：用 51 位二进制数来表示，其中的“0”“1”作为基因；51 位分别表示数据库每行的 51 个点，第 n 位代表第 n 个点。“0”代表改组取点不包含这个点，反之“1”代表包含此点。在这种规则下，第 1 位和第 51 位必须是 1.

完成上述“基因”构造后，初始化为随机设置 0-1 序列。

2.杂交

此过程模仿了自然界的染色体杂交过程：在染色体联会时，非姐妹染色单体有一定概率发生交叉，交换部分基因序列。交叉位置可能在染色体上的任意位置。

我们选取了单点交换的方法：

随机选择两条染色体，有一定概率的随机选取基因交叉的位置（51 个数据点共 50 个交换位置）并且将交叉后的部分相互交换。

3.变异

由于采用了二进制编码的表示方法，基因突变的模仿方式非常简单：设置一个突变概率 t（设置得过小进化速度过慢，设置的过大容易变成一个随机突变不服从进化规律的族群），每一位上的编码按照概率 t 突变成反码。即“0”变成“1”，“1”变成“0”。此过程中头尾两位同样不能变异。

4.自然选择

模仿自然选择过程，关键在于适应度函数以及选择函数的定义是否能够准确反应基因的“优越性”。

4.1 适应度函数的定义：

达尔文进化论中的重要理论“物竞天择，适者生存”是生物进化的源动力，其根源在于，生物在于外界环境抗争的过程中强者才有更大的概率活下来，于是好的“基因”更有可能在“后代”的身上得到体现。我们的任务在于如何让适应度衡量该点的生存概率。

本次任务的具体实现方案中，我们直接设置拟合成本的倒数为适应度，因为本体需要我们寻找成本最小的取点方案，倒数能够真实的反映这一情况。

4.2 选择函数的定义

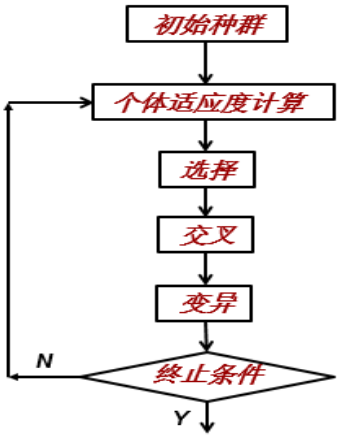
自然界中，更能适应环境的个体并非一定能够留下更多的后代，而是“有更大的概率来留下后代”。

为了真实反映这一选择过程，我们采用了“轮盘赌的方法。既将个体适应度多占总数的百分比反映在圆形轮盘上的区域，生成随机数，等概率的落到圆盘上的任一个角度上，所落区域代表该区域留下后代。数学实现上，将 1 按照比例划分，生成随机数落到不同的区间代表生成相应的后代，落下的次数越多说明留下的后代越多。

5.终止条件

提前设置迭代次数，到达次数终点时终止即可。

6.算法的基本流程可以用下面的流程图来表示。



（2）算法的具体实现过程

这里罗列一些关键算法的实现，具体实现过程的代码见附录。本次试验将每一个样本中取 7 个点作为一个生物个体，并且计算其适应度。其中一些参变量的设置如下：

popSize: 代表种群大小；
pop: 代表种群；
cost：代表适应度；
popration: 代表进化代数为 50；
p_crs: 代表交叉概率为 0.90；
p_mutation: 代表变异概率为 0.001；

以下为算法的实现过程：

1. 读入样本数据。

2. 用随机函数随机选取 7 个点得到一个数组，对这个数组中的元素进行排列，作为个体基因的染色体。

3. 利用初始化种群函数 creation(popSize)随机生成 popSize 个生物个体的染色体，存在种群列表 pop 中作为初始种群。

4. 个体适应度函数 adapt(pop,y,popSize)计算个体的平均成本作为其适应度。函数中内嵌函数 errorcost(dy)使用式 3-1 所规定的的误差计算方案计算单个个体误差成本。将所计算的适应度存放在适应度列表 cost 中。

5. 自然选择函数 select(pop,cost,popSize)根据适应度的要求选择能繁殖后代的个体，储存进 pop 中作为下一代的父代。因为自然选择所保留的生物个体是在自然中适应度较高的那部分，因此设置该个体 i 被选择的概率为：

$$p_i = \frac{\text{第}i\text{个个体的适应度}}{\text{总适应度}}$$

将每个个体被选择的概率存在一个列表 s 中，然后使用二分法查找函数找到适应度高的个体存在新的 pop 中，作为新的种群，即为下一代的父代。

6. 交叉过程函数 mutationpop(pop,popSize,p_crs)。在选择出种群父代后，在其内进行交叉。我们采用单点交叉的方式，在基因段中随机一处断裂并与另一基因进行重组，设定交叉概率 p_crs=0.85。将父代随机排列并按照 i 和(popSize-i+2)进行配对，其中 popSize=100。若二者进行交叉，则随机产生一个断裂点，将二者基因重组后加入新的种群；若二者不进行交叉，则将二者输入新的种群。

7. 变异过程函数 mutate(pop,popSize,p_mutation)即将基因中的碱基进行随机替换。由于变异概率较小，且仅仅提供基因的多样性，所以我们设定变异的概率 p_mutation=0.001。由于种群中

每一个个体都可能发生突变，且种群是由二维矩阵表示的，因此变异过程需要使用二重循环来实现。

8.主函数按照遗传算法的基本流程，先调用样本数据，然后使用初始化种群函数creation(popSize)生成初始种群，然后进入第一代循环；在每一代循环中，先调用个体适应度函数对种群个体的适应度进行计算，然后调用自然选择函数选择出适应度较高的个体作为下一代个体的父代，并进行交叉和变异，之后判断是否达到预设的代数，若未达到，则再次进入相同的循环，这次对象即为上次循环所生成的父代。每次循环都会调用 display 函数来输出结果，便于观察。

七 实验结果

1.以三次多项式拟合得到的最优选点

[1、14、27 、38、51]，其成本为122.79。

2.以三次样条插值得到的最优选点

[1、10、21、31、43、51]，其成本为95.41。

对比之下，我们选择以三次样条插值为插值方法，点[1,9,19,26,33,44,51]，作为选点的定标方案。

参考文献

[1] “统计推断”课程设计的要求 V2.2 2015-9-22

[2] “统计推断”课程设计的要求 V2.2 2015-9-22

[3] 维基百科. poptic_algorithm[EB/OL]
https://en.m.wikipedia.org/wiki/poptic_algorithm

[4] 统计推断讲座 2_问题的提出和基本求解思路

附录

暴力算法代码

```
warning off all;
clear;
clc;
tic;
alldata = xlsread('data.xls','Sheet1','A1:AY938','basic');
Min = 1026;
    for p1 = 1:45
        for p2 = (p1+1):46
            for p3 = (p2+1):47
                for p4 = (p3+1):48
                    for p5 = (p4+1):49
                        for p6 = (p5+1):50
                            for p7 = (p5+1):51
                                Points = [p1 p2 p3 p4 p5 p6 p7];
                                Scores1=curvefits1(alldata,Points)
                                if Scores1< Min
                                Min = Scores1;
                                disp(Points);
                                disp(Min);
                            end;
                        end;
                    end;
                end;
            end;
        end;
    end;
end;
toc;
```

遗传算法代码

```
clear all;
clc;
p_crs = 0.9;
p_mutation = 0.08;
pop_size = 100;
alldata = xlsread('data.xls');
k = 0;
while ( k ==0 )
[pop,k]= creation(pop_size);
end;
loop = 50;
Points = zeros(1,7);
tic;
for t = 1:loop
new_pop1 = mutationpop(pop ,pop_size, p_mutation,alldata);
new_pop = cross(new_pop1,alldata,p_crs);
pop = new_pop;
disp(['popration',num2str(t),':']);
disp(new_pop);
min=new_pop(1,7);
for r = 1:pop_size
if (new_pop(r,7) < min)
min= new_pop(r,7);
Points = new_pop(r,1:7);
end;
end;

disp(Points);
end;
toc;
```

```
function [ new_pop ] = cross(pop,alldata,p_crs)
[p_x p_y] = size(pop);
new_pop = pop;
for i=1:2:(p_x-1)
    if(rand<p_crs)
        C_point = randi([2,6]);
        score1 = pop(i,7)+pop(i+1,7);

while((pop(i,C_point-1)==pop(i+1,C_point))||(pop(i+1,C_point-1)==pop(i,C_
point)))
```

```
            C_point = mod((C_point-1),5)+2;
            %disp('tiaozheng');
        end;
        new_pop(i,1:6) = [pop(i,1:C_point-1) pop(i+1,C_point:(p_y-1))];
        new_pop(i+1,1:6) = [pop(i+1,1:C_point-1) pop(i,C_point:(p_y-1))];
        new_pop(i,7) = curvefits(alldata,new_pop(i,1:6));
        new_pop(i+1,7) = curvefits(alldata,new_pop(i+1,1:6));
        score2 = new_pop(i,7)+new_pop(i+1,7);
        if(score2>score1)
            new_pop(i,:)=pop(i,:);
            new_pop(i+1,:)=pop(i+1,:);
        end;
    end;
end;
end;
```

```
function [pop k] =creation(pop_size)
k = 0;
pop = ones(pop_size,7)*100;
for i = 1:pop_size
    pop(i,1) = randi([1,5]);
    pop(i,2) = randi([6,10]);
    pop(i,3) = randi([11,18]);
    pop(i,4) = randi([19,30]);
    pop(i,5) = randi([31,40]);
    pop(i,6) = randi([41,50]);
end;

for t = 1:pop_size-1
    for m = (t+1) : pop_size
        for n = 1:6
            if (pop(t,n)~= pop(m,n))
                k = 1;
            break;
            end;
        end;
    end;
end;
end
```

```
function Score = curvefits(alldata,points)
Xi = zeros(1,51);
Score = 0;
for i=1:400
    PointX = [alldata(2*i-1,points(1)) alldata(2*i-1,points(2))
alldata(2*i-1,points(3)) alldata(2*i-1,points(4)) alldata(2*i-1,points(5))
alldata(2*i-1,points(6)) ];
    PointY = [alldata(2*i,points(1)) alldata(2*i,points(2))
alldata(2*i,points(3)) alldata(2*i,points(4)) alldata(2*i,points(5))
alldata(2*i,points(6)) ];
for m = 1:51
    Xi(m) = alldata(2*i-1,m);
end;
y = interp1(PointX,PointY,Xi,'spline');
Diff = y - alldata(2*i,:);
score = 72;
for t=1:51
    if abs(Diff(t))<=0.5
        score=score+0;
    elseif abs(Diff(t))<=1
        score=score+0.5;
    elseif abs(Diff(t))<=2
        score=score+1.5;
    elseif abs(Diff(t))<=3
        score=score+6;
    elseif abs(Diff(t))<=5
        score=score+12;
    else score=score+25;
end;
end;
Score = Score+score;
end;
Score = Score/100;
```

多项式拟合功能函数

```
function [ Scores1 ] = curvefits1(alldata,points)
Scores1 = 0;
for i=1:469
```

```
    x = [alldata(2*i-1,points(1)) alldata(2*i-1,points(2))
    alldata(2*i-1,points(3)) alldata(2*i-1,points(4)) alldata(2*i-
    1,points(5)) alldata(2*i-1,points(6)) alldata(2*i-1,points(7))];
    z = [alldata(2*i,points(1)) alldata(2*i,points(2))
    alldata(2*i,points(3)) alldata(2*i,points(4))
    alldata(2*i,points(5)) alldata(2*i,points(6))
    alldata(2*i,points(7))];
    p3 =polyfit(x,z,3);
    y = polyval(p3,alldata(2*i-1,:));
    Diff = y - alldata(2*i,:);
    score = 12*7;
    for t=1:51
        if abs(Diff(t))<=0.5
        score=score+0;
        elseif abs(Diff(t))<=1
        score=score+0.5;
        elseif abs(Diff(t))<=2
        score=score+1.5;
        elseif abs(Diff(t))<=3
        score=score+6;
        elseif abs(Diff(t))<=5
        score=score+12;
        elseif abs(Diff(t))>5
        score=score+25; end;
    end;
    Scores1 = Scores1 +score;
end;
Scores1 = Scores1/469;
```

三次样条插值拟合函数

```
function Scores = curvefits(alldata,points)
Xi = ones(1,51);
Scores = 0;
for i=1:469
        PointX = [alldata(2*i-1,points(1)) alldata(2*i-1,points(2))
    alldata(2*i-1,points(3)) alldata(2*i-
        1,points(4)) alldata(2*i-1,points(5)) alldata(2*i-1,points(6))
    alldata(2*i-1,points(7))];
        PointY = [alldata(2*i,points(1)) alldata(2*i,points(2))
    alldata(2*i,points(3))
        alldata(2*i,points(4)) alldata(2*i,points(5))
    alldata(2*i,points(6)) alldata(2*i,points(7))];
    for m = 1:51
        Xi(m) = alldata(2*i-1,m);
```

```
    end;
        y = interp1(PointX,PointY,Xi,'spline');
        Diff = y - alldata(2*i,:);
    score = 12*7;
    for t=1:51
        if abs(Diff(t))<=0.5
        score=score+0;
        elseif abs(Diff(t))<=1
            score=score+0.5;
        elseif abs(Diff(t))<=2
        score=score+1.5;
        elseif abs(Diff(t))<=3
        score=score+6;
        elseif abs(Diff(t))<=5
        score=score+12;
        elseif abs(Diff(t))>5
        score=score+25;
        end;
    end;
        Scores = Scores + score;
    end;
Scores = Scores/469;
```