统计推断在数模转换系统中的应用

66 组 詹诗渊 5140309157 尤梓荃 5140309043

摘要： 本课题依托于 MATLAB 工具程序语言进行计算机辅助分析，通过对于遗传算法以及模拟退火算法的研究，为某投入试生产的电子产品一内部模块的批量生产设计一种合理的传感特性校准方案，期间需要建立数学模型研究传感部件特性，并进行相应成本计算。本文阐述了研究项目评估成本的具体过程。

关键词： 多项式拟合 三次样式插值 模拟退火算法 遗传算法

Application of Statistical Inference in DA Inverting System

No.66   Zhan Shiyuan 5140309157   You Ziquan 5140309043

ABSTRACT: Based on MATLAB as an instrument to analysis the project has designed an reasonable sensing characteristic calibration scheme for the manufacture of a pre-producing module in an electronic product. Meanwhile we need to establish mathematical models and cost calculation. This article illustrate the processing of the cost evaluation.

Key words： Polynomial fitting; Cubic spline interpolation; Simulated annealing algorithm; Genetic algorithm

1 引言

统计推断（statistical inference） 根据带随机性的观测数据（样本）以及问题的条件和假定（模型），而对未知事物作出的，以概率形式表述的推断。本次课题研究的是假定有某型号即将投放市场的电子产品，其中有一个电子模块，其主要作用是检测外部环境的各种物理量。其使用的传感器部件输入输出特性呈现出明显的非线性。本课题通过建立数学

模型，针对传感器部件进行标定，且过程尽量简单，成本尽量低。

2 模型假设与变量建立

2.1 模型

对于本课题的问题研究需要对该电子模块建立数学模型并进行相关分析。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

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

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

2.2 传感部件特性

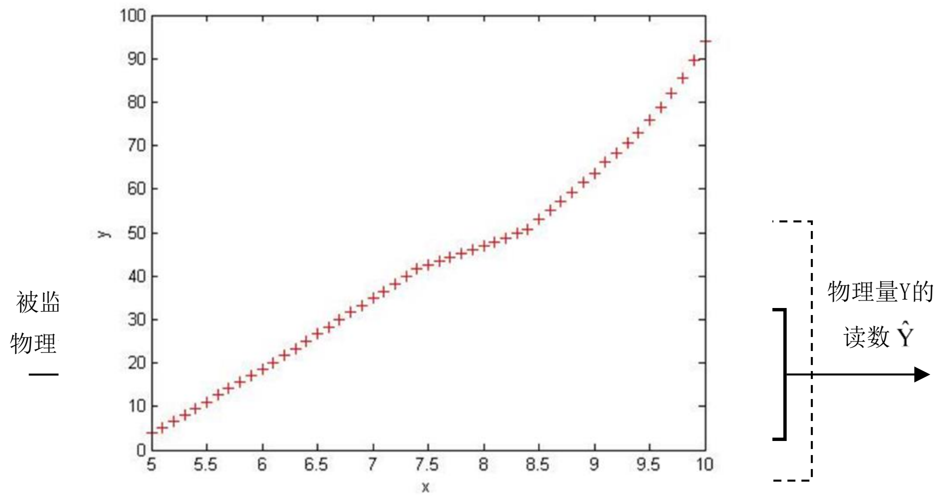

图 2-2-1 模型展示

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。

下面四幅图是四个样本数据点图像：

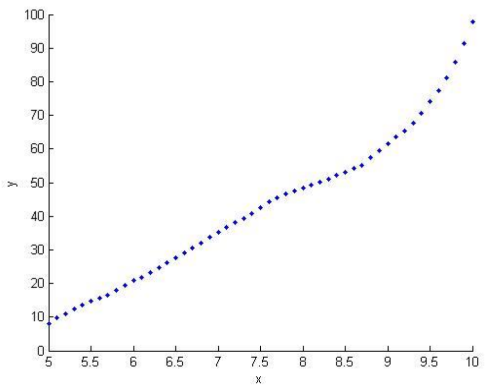

图 2-2-2 第 20 组数据

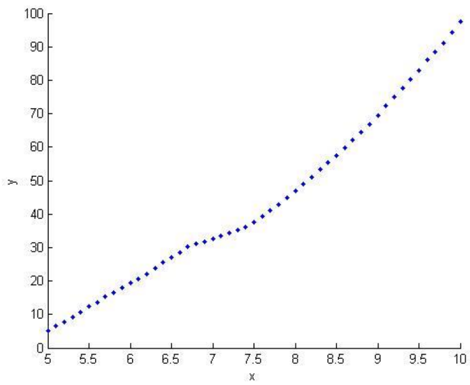

图 2-2-3 第 128 组数据

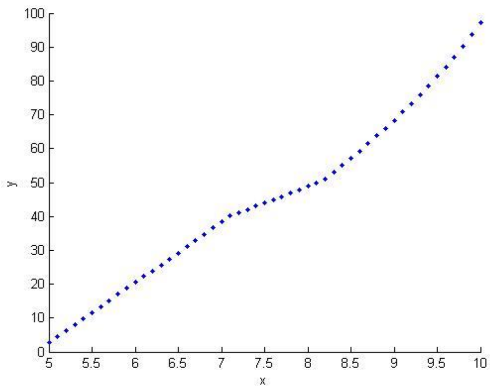

图 2-2-4 第 284 组数据

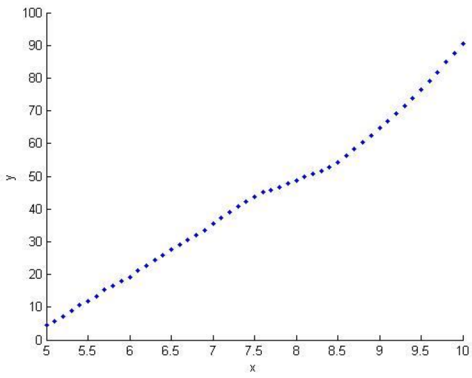

图 2-2-5 第 392 组数据

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。

7　单点定标误差成本

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

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。

8　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。

9　某一样本个体的定标成本

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

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。

10　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。

$$C=\frac{1}{M}\sum_{i=1}^{M}S_i \tag{3}$$

总成本较低的校准方案，认定为较优方案。

成本函数见附录。

3　主要数值计算方法

3.1 拟合方式选取

在数值分析中，插值过程可以具体使用线性插值，三次样条插值，立方插值等操作。在本次项目中，我们小组将主要讨论三次样条插与分段 3 次 Hermite 插值多项式以及多项式拟合。根据后文的数据比较，我们小组最终选择的操作是分段 3 次 Hermite 插值多项式。

3.1.1 多项式拟合

数据拟合的具体作法是：对给定数据 (x_i, y_i) (i=0, 1, …, m)，在取定的函数类 Φ 中,求 $p(x) \in \Phi$,使误差 $r_i = p(x_i) - y_i$ (i=0, 1, …, m)的平方和最小，即

$$\sum_{i=0}^{m} r_i^2 = \sum_{i=0}^{m} \left[p(x_i) - y_i\right]^2 = \min$$

从几何意义上讲，就是寻求与给定点 (x_i, y_i) (i=0, 1, …, m)的距离平方和为最小的曲线 $y = p(x)$ （图 6-1）。函数 $p(x)$ 称为拟合 函数或最小二乘解，求拟合函数 $p(x)$ 的方法称为曲线拟合的最小二乘法。

在曲线拟合中，函数类 Φ 可有不同的选取方法.假设给定数据点 (x_i, y_i) (i=0, 1, …, m)， Φ 为所有次数不超过 $n(n \le m)$ 的多项式构成的函数类，现求一 $p_n(x) = \sum_{k=0}^{n} a_k x^k \in \Phi$,使得

$$I = \sum_{i=0}^{m} \left[p_n(x_i) - y_i\right]^2 = \sum_{i=0}^{m} \left(\sum_{k=0}^{n} a_k x_i^k - y_i\right)^2 = \min \qquad (1)$$

当拟合函数为多项式时，称为多项式拟合，满足式（1）的 $p_n(x)$ 称为最小二乘拟合多项式。特别地，当 n=1 时，称为线性拟合或直线拟合。

此处，我们小组分别采用了三次拟合，四次拟合，五次拟合，得到的数据如下图所示。选取的特征点为运用模拟退火算法以及遗传算法初步选定的六组特征点，分别为(1)[1,7,12,24,33,48,51] (2)[1,6,13,25,36,47,51] (3)[1,4,14,24,37,49,51] (4)[1,4,17,29,38,47,51] (5)[1,5,16,29,38,48,51] (6)[1,5,14,27,39,45,51]，六组数据点并非最后优选点，仅用于比较优劣。

多项式次数	平均残差平方和
三次拟合	67.8933
四次拟合	49.4626
五次拟合	41.1857

表 3-1-1 多项式拟合所得平均残差平方和

由表中的数据可以看出，平均残差平方和随次数增加而减少，但即便是五次拟合误差还是比较大。

3.1.2 三次样式插值法及分段 3 次 Hermite 插值多项式

所谓三次样条插值多项式 $S_n(x)$ 是一种分段函数，它在节点 x_i $(a = x_0 < x_1 < \cdots < x_{n-1} < x_n = b)$ 分成的每个小区间 $[x_{i-1}, x_i]$ 上是 3 次多项式，其在此区间上的表达式如下：

$$S(x) = \frac{1}{6h_i}[(x_i - x)^3 M_{i-1} + (x - x_{i-1})^3 M_i] + (y_{i-1} - \frac{h_i^2}{6} M_{i-1}) \frac{x_i - x}{h_i} + (y_i - \frac{h_i^2}{6} M_i) \frac{x - x_{i-1}}{h_i},$$
$$x \in [x_{i-1}, x_i], i = 1, 2, \cdots, n.$$

因此，只要确定了 M_i 的值，就确定了整个表达式， M_i 的计算方法如下：

令：

$$\begin{cases} \mu_i = \dfrac{h_i}{h_i + h_{i+1}}, \lambda_i = \dfrac{h_{i+1}}{h_i + h_{i+1}} = 1 - \mu_i \\ d_i = \dfrac{6}{h_i + h_{i+1}} (\dfrac{y_{i+1} - y_i}{h_{i+1}} - \dfrac{y_i - y_{i-1}}{h_i}) = 6 f(x_{i-1}, x_i, x_{i+1}) \end{cases}$$

则 M_i 满足如下 $n-1$ 个方程：

$$\mu_i M_{i-1} + 2M_i + \lambda_i M_{i+1} = d_i, i = 1, 2, \cdots, n-1$$

对于第一种边界条件下有

$$\begin{cases} M_{n-1} + 2M_n = \dfrac{6(f'_n - f[x_{n-1}, x_n])}{h_{n-1}} \\ 2M_0 + M_1 = \dfrac{6([x_1, x_0] - f'_0)}{h_0} \end{cases}$$

如果令 $\lambda_0 = 1, d_0 = \dfrac{6(f[x_1, x_0] - f'_0)}{h_0}, \mu_n = 1, d_n = \dfrac{6(f'_n - f[x_{n-1}, x_n])}{h_{n-1}}$，那么解就可以为

$$\begin{pmatrix} 2 & \lambda_0 & & & \\ \mu_1 & 2 & \lambda_1 & & \\ & \ddots & \ddots & \ddots & \\ & & \mu_{n-1} & 2 & \lambda_{n-1} \\ & & & \mu_n & 2 \end{pmatrix} \begin{pmatrix} M_0 \\ M_1 \\ \\ M_{n-1} \\ M_n \end{pmatrix} = \begin{pmatrix} d_0 \\ d_1 \\ \vdots \\ d_{n-1} \\ d_n \end{pmatrix}$$

分段三次 Hermite 插值的数学模型为：

$$H(x) = \begin{cases} H_1(x), x \in [x_0, x_1] \\ H_2(x), x \in [x_1, x_2] \\ \cdots\cdots \\ H_n(x), x \in [x_{n-1}, x_n] \end{cases}$$

$$H_i=\frac{[h_i+2(x-x_{i-1})](x-x_i)^2}{h_i^3}y_{i-1}+\frac{[h_i+2(x-x_i)](x-x_{i-1})^2}{h_i^3}y_i$$

$$+\frac{(x-x_{i-1})(x-x_i)^2}{h_i^2}y_{i-1}^{'}+\frac{(x-x_{i-1})^2(x-x_i)}{h_i^2}y_i^{'}$$

同样的，我们对之前选定原来的六组点分别为(1)[1,7,12,24,33,48,51] (2)[1,6,13,25,36,47,51] (3)[1,4,14,24,37,49,51] (4)[1,4,17,29,38,47,51] (5)[1,5,16,29,38,48,51] (6)[1,5,14,27,39,45,51]，得到的数据如下图所示。

插值方式	平均残差平方和
三次样条插值	28.9346
分段 3 次 Hermite 插值多项式	28.4919

表 3-1-2 三次样条插值拟合及分段 3 次 Hermite 插值多项式拟合所得平均残差平方和

与表 3-1-1 相比，可以看出，三次样条插值拟合及分段 3 次 Hermite 插值多项式拟合相比于各类多项式拟合平均残差平方和要小得多，即误差要小得多。所以经比较，我们决定采用分段 3 次 Hermite 插值多项式拟合。

3.2 模拟退火算法

模拟退火（SA）是一种基于 Monte Carlo 迭代法的启发式随机搜索算法。 SA 来源于对固体物质的退火降温过程中的热平衡问题的模拟和随机搜索优化问题，以找到全局最优解或近似全局最优解[5]。SA 在寻找最优解的过程中，不仅接受优解，而且根据随机的验收标准，在一定程度上接受恶化解(Metropolis 准则)。此外，接受恶化解的概率逐渐趋向于 0，这使得能够跳出局部极值区，从而找到全局最优解，所以要确保算法的收敛性。

程序设计思路如下：

（1）初始化：设置初始温度T_0=100，温度衰减系数godown=0.995，终止温度设定为t=5，在每个T值中在设置一个循环次数L=20（增加循环次数，相当于微调）。初始随机生成一组解。

（2）对于每一个降低到的温度，迭代若干次：每次循环中，新解与原解相比只有一组数据变化，其余数据不变。

（3）求原解和新解的能量值，扰动求新解，比较原解和新解对应的“能量”值，如果新解“能量”值低，即成本低，则采取新解，否则以一定的小概率接受新解。

（4）若达到一定的次数没有接受新解则跳出算法，直接结束。否则迭代完次数后降低温度，回到步骤（3），直到降低到最小的温度为止。

在模拟退火算法中，当新解的成本比最优解的成本大时，需要以一定概率接受新解为最 优解，这是防止陷入局部最优解。这里的概率公式的选取很关键，既要保证跳变概率不大， 又不能太小（影响算法速度），经过多次试验，我们选取了 1/[10*t*(newCost-minimumCost)]。

最后的得到的数据表如下图所示：

取点个数	特征值组合	校准方案总体	实验耗时/s

		成本	
4	46 20 33 5	101.63	569.3
5	23 4 15 35 50	89.33	582.0
6	35 22 27 4 49 13	89.24	571.4
7	3 13 51 18 45 33 26	94.44	571.0
8	44 34 28 2 21 14 40 51	103.03	570.7
9	32 50 2 25 36 45 8 15 21	112.64	570.4
10	28 34 20 25 17 10 45 49 2 41	123.57	568.5

表 3-2-1 模拟退火算法所得特征值组合及相关数据

3.3 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。

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=100。初始染色体为 51 位的二进制串 000....00，取点个数 k 时，将特征值点所对应的位置上的值置为 1。我们设定初始种群数量为 200。

b)个体评价：对每一个特征值串进行拟合，求得成本后，将成本作为个体适应度。平均成本越小，个体适应度越大。

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。通过适应度函数计算每个个体的适应度，在[0,1]内为他们分配区间，区间长度与适应度成正相关，再随机生成[0,1]内的随机数，该随机数所在区间所对应的的个体遗传到下一代。

d)交叉运算：以一定概率 p=0.25 来进行交叉。随机选取两个个体，然后随机选取特征值串的一个位置，将此位置之后的片段进行交换，完成交叉过程。

e)变异运算：以一定概率 p 来进行变异，在保证个体染色体长度不变的情况下，我们先生成随机数 $r_1 \in \{a_1,a_2,...a_k\}$,将 r_1 所对应的位置上的值置为 0，再生成随机数 r_2 不属于 $\{a_1,a_2,...a_k\}$，将 r_2 所对应位置上的值置为 1.

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。

编写成代码后经 MATLAB 计算，得到数据表如图：

取点个数	特征值组合	校准方案总体成本	实验耗时/s
4	5 21 33 48	99.8	3387
5	4 17 27 35 47	86.3	3374
6	3 16 25 32 41 48	89.1	3403
7	2 15 21 28 29 41 49	97.8	3346
8	5 12 22 26 32 39 45 51	103.0	3303
9	31 2 19 46 11 15 37 26 50	113.2	3612
10	3 10 20 23 27 30 33 38 44 50	123.4	3430

表 3-3-1 遗传算法所得特征值组合及相关数据

与表 3-2-1 相比可以看出遗传算法相比较模拟退火算法是一种牺牲实验耗时换取校准方案总体成本的算法，适合于精确地拟合。

4 结果分析

经过以上的实验探究，我们分别针对不同的拟合方式以及特征值选取进行了挖掘探讨。针对拟合方式，多项式拟合相比于三次样式插值法及分段 3 次 Hermite 插值法，有着误差较大的缺点，权衡之下我们选择了分段 3 次 Hermite 插值法。而对于特征值的选取，我们采用了模拟退火算法以及遗传算法，比较之后发现，模拟退火算法拥有精确度稍低而时间较快的特点，遗传算法具有精确度高而耗时稍长的特点，针对我们的数据样本大小以及探究要求，最后我们决定选择遗传算法作为主要探究方法。

最后的实验结果如下：

取点个数	特征值组合	校准方案总体成本	实验耗时/s
5	4 17 27 35 47	86.3	3374

5 参考资料

[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]维基百科. 曲线拟合-维基百科，自由的百科全书[M/OL].[2014-01-14]. http://zh.wikipedia.net.ru/wiki/曲线拟合

[3]维基百科. 样条函数-维基百科，自由的百科全书[M/OL].[2015-01-18]. http://zh.wikipedia.net.ru/wiki/样条函数

[4]维基百科. 模拟退火-维基百科，自由的百科全书[M/OL].[2015-04-14]. http://zh.wikipedia.net.ru/wiki/模拟退火

[5]维基百科. 遗传算法-维基百科，自由的百科全书[M/OL].[2015-10-11]. http://zh.wikipedia.net.ru/wiki/遗传算法

6 附录

老师给出的成本函数：test_ur_answer

```
%%%%%%% 答案检验程序 2015-11-04 %%%%%%%%

my_answer=[4,17,27,35,47];%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;

% 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];

% 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
```

小组合作写出的成本函数:

```
    function y=cost(answer)
    %成本函数
    %    此处显示详细说明
    answerSize=length(answer);
    data=load('20150915dataform.csv');
    [m,n]=size(data);
    x=zeros(m/2,n);
    y0=zeros(m/2,n);
    y1=zeros(m/2,n);
    for i=1:m/2
        x(i,:)=data(2*i-1,:);
        y0(i,:)=data(2*i, : );
    end
    answer_gene=zeros(1,n);
    answer_gene(answer)=1;
    index_temp=logical(answer_gene);
    x_choose=x(:,index_temp);
    y_choose=y0(:,index_temp);
    for i=1:m/2
        y1(i,:)=curve(x_choose(i,:),y_choose(i,:));
    end
    q=12;
    erro=abs(y1-y0);
    cost_0_1=(erro>0.4&erro<=0.6);
    cost_0_7=(erro>0.6&erro<=0.8);
    cost_0_9=(erro>0.8&erro<=1.0);
    cost_1_5=(erro>1.0&erro<=2.0);
    cost_6=(erro>2.0&erro<=3.0);
    cost_12=(erro>3.0&erro<=5.0);
    cost_25=(erro>5.0);
    s=0.1*cost_0_1+0.7*cost_0_7+0.9*cost_0_9+1.5*cost_1_5+6*cost_6+12
    *cost_12+25*cost_25;
    si=sum(s,2)+q*ones(m/2,1)*answerSize;
    y=sum(si)/(m/2);
```

拟合函数:

```
function y=curve(x_data,y_data)
% 插值拟合
```

```
x=[5:0.1:10];
y=interp1(x_data,y_data,x,'pchip');
end

function y = curve1( x_data,y_data )
%   多项式拟合
fit=polyfit(x_data,y_data,5);
x=[5:0.1:10];
y=polyval(fit,x);
end
```

测试拟合方式的优劣：Test

```
data=csvread('20150915dataform.csv');
[m,n]=size(data);
x=zeros(m/2,n);
y0=zeros(m/2,n);
y1=zeros(m/2,n);
for i=1:m/2
    x(i,:)=data(2*i-1,:);
    y0(i,:)=data(2*i,:);
end
test=[1,7,12,24,33,48,51; 1,6,13,25,36,47,51; 1,4,14,24,37,49,51;
1,4,17,29,38,47,51; 1,5,16,29,38,48,51; 1,5,14,27,39,45,51];
s=0;
for k=1:6
    answer_gene=zeros(1,n);
    answer_gene(test(k,:))=1;
    index_temp=logical(answer_gene);
    x_choose=x(:,index_temp);
    y_choose=y0(:,index_temp);
    for i=1:m/2
        y1(i,:)=curve1(x_choose(i,:),y_choose(i,:));
```

%上面测试的是多项式拟合，当要测试插值拟合时，只要把 curve1 改成 curve 即可。而多项式拟合只要把 curve1 里的 polyfit(x_data,y_data,n)分别代入 3,4,5 即可得出三，四，五次拟合的结果。插值拟合只要把 curve 函数 y=interp1(x_data,y_data,x,'method')的 method 换成‘spline’与’pchip’就能得到两种插值方式的比较。

```
    end
    r=y1-y0;
    r=r.^2;
    s=s+sum(r(:))/2400;
end
```

heredity.m

```
tic;
rng(100);
sampleSize=6;%这里的数字可变，‘6’只是示例
popSize=200;
pop=zeros(popSize,sampleSize);
for i=1:popSize
    pop(i,:)=randperm(51,sampleSize);
end
cost_m=zeros(1,popSize);
for i=1:popSize
   cost_m(1,i)=cost(pop(i,:));
end
for generation=1:100;
    selection;
    crossover;
    mutationRate=0.4;
        for i=1:popSize
         r=rand();
           if (mutationRate>r)
               pop(i,:)=mutation(pop(i,:));
           end
        end

        for i=1:popSize
            cost_m(1,i)=cost(pop(i,:));
        end
        minpop=min(cost_m);
        fprintf('%2d %5.1f 5.1f',generation,mean(cost_m),min(cost_m))
    ;
        for i=1:popSize
            if (cost_m(1.i)==min(cost_m))
                for j=1:sampleSize
                    fprintf('%2d   ',pop(i,j));
                end
                fprintf('\n');
                break;
            end
        end
    end
    toc;
```

```
%selection.m
%选择算子，selection 的算法
adapt_temp=-cost_m+max(cost_m)+1;
adapt_rate=adapt_temp/sum(adapt_temp);
adapt=zeros(1,popSize);
for i=1:popSize
    adapt(i)=sum(adapt_rate(1:i));
end
tempPopSize=popSize;
tempPop=zeros(tempPopSize,sampleSize);
pmin=min(cost_m);
for i=1:popSize
    if(cost_m(i)==pmin)
        break
    end
end
tempPop(1,:)=pop(i,:);
tempPop(popSize/2,:)=pop(i,:);
tempPop(popSize,:)=pop(i,:);
for i=[2:tempPopSize/2-1,(tempPopSize/2+1):(tempPopSize-1)]
    r=rand();
    for j=1:popSize
        if(adapt(j)>=r)
            break;
        end
    end
    tempPop(i,:)=pop(i,:);
end
pop=tempPop;
popSize=tempPopSize;
```

```
crossover.m
%交换算子，crossover 的算法
p0=0.25;
flag=0;
times=10;
a=0;
b=0;
for i=1:times
    for k=1:popSize
        r=rand();
        if(r<p0&&flag==0)
            a=k;
            flag=1;
```

```
        end
        r=rand();
        if(r<p0&&flag==1)
            b=k;
            flag=2;
            break;
        end
    end
    if(flag==2&&a~=b)
    r=randi([2 sampleSize],1);
    temp=pop(a,r:sampleSize);
    pop(a,r:sampleSize)=pop(b,r:sampleSize);
    pop(b,r:sampleSize)=temp;
    end
end
```

mutation.m

```
function newanswer =mutation(answer)
%变异算子
n=51;
answer_gene=zeros(1,n);
answer_gene(answer)=1;
index=randi(n,1);
if(ismember(index,answer))
    answer_gene(index)=0;
    flag=1;
else
    answer_gene(index)=1;
    flag=0;
end

index=randi(n,1);
if(flag)
    while(ismember(index,answer))
        index=randi(n,1);
    end
    answer_gene(index)=1;
else
    while(~ismember(index,answer))
        index=randi(n,1);
    end
    answer_gene(index)=0;
end
point=1:51;
```

```
newanswer=point(logical(answer_gene));
end
```

退火算法:

```
    tic;
    rng(100);
    t=100;
    turn=0;%循环变量，想观察循环的次数
    godown=0.99;
    sampleSize=6;%同理，这里的样本数可改变，'6'只是示例
    answer=randperm(51,sampleSize);
    newAnswer=answer;
    minimumCost=cost(answer);
    L=20;
    while(t>1)
        for i=1:L
            index=randperm(sampleSize,1);
            change=mod(newAnswer(index),51)+1;
            while(ismember(newAnswer,change))
                change=mod(change,51)+1;
            end
            newAnswer(index)=change;
            newCost=cost(newAnswer);
            r=rand();
            if(newCost<=minimumCost)
                minimumCost=newCost;
                answer=newAnswer;
            end
            turn=turn+1;
        end
        t=t*godown;
    end
    for j=1:sampleSize
        fprintf('%2d ',answer(j));
    end
    fprintf('\n cost is %3.2f \n',minimumCost);
toc;
```