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 65 组，成员黄睿轩　学号 5137319002　　马胤磊　学号 5140309175 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
退火算法	《统计推断在数模模数转换中的应用》，孙煜，2014 年秋季学期，组号 50 参考代码所用方法及其思路。
结果统计和分析	《统计推断在数模转换系统中的应用》，艾玥祺，2014 年秋季学期，组号 20 最优的选点方式。

统计推断在数模转换系统中的应用

第 65 组　　黄睿轩　5137319002　　马胤磊　5140309175

摘要：本文的目标是为具有传感效应的模块的批量生产设计一种成本合理的校准（定标工序）方案。本方案以 MATLAB 为工具，运用模拟退火算法统计数据，选取特征点，并三次样条插值法等多种数据运算求解方法来拟合出几种曲线，进而进行单点到总体的成本分析和比较，从而选取出最优的拟合方案来确定传感特性的校准方案。

关键词：　模拟退火，曲线拟合，样条插值，方案优化

Application of Statistical Inference in DA Inverting System

Abstract: The objective of this paper is to design a reasonable cost standard (calibration process) program for sensor module. The program using MATLAB as a tool, and using simulated annealing algorithm statistics, select the feature point; three times spline interpolation method for solving arithmetic method to fit out several curves. Finally, compare the overall cost, thereby selecting the optimal fitting program to determine the characteristics of the sensor calibration scheme.

Keyword: Simulated Annealing , curve fitting，Cubic Spline Interpolation, Scheme Optimization

1.引言

传感特性校准，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（传感器部件检测的对象物理量）与 X 值（传感部件的输出电压信号）间一一对应的特性关系的过程。如下数学模型能够获得信号 $\hat{Y}$ 作为 Y 的读数（监测模块对 Y 的估测值），再通过数学方法确定适用于该个体的估测函数 $\hat{Y}$=f（x）。

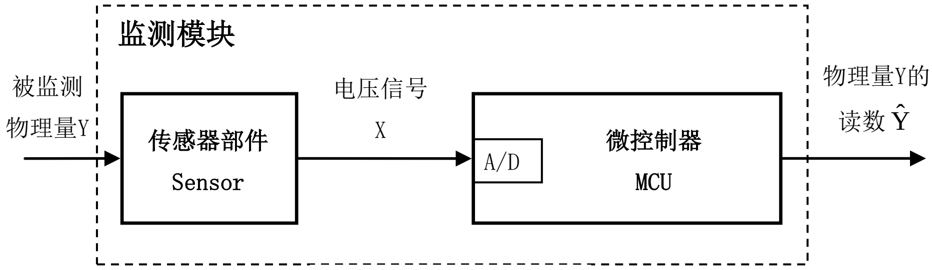


图 1.检测模块组成

2.课程的基本问题

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方案。

由于所给实验数据容量较大，不易于处理和分析，所以就要求对传感器检测的数据进行随机的抽样和分组，经过成本分析来设计一种成本最优的校准方案。

3 样本数据分析

通过分析如下图所示的不同样本特性的图示对比，我们可以总结出传感器输入输出的主要的特性

（1）Y 取值随 X 取值的增大而单调递增；

（2）不同个体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，它们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；

（4）不同个体的中段起点位置、终点位置有随机性差异。

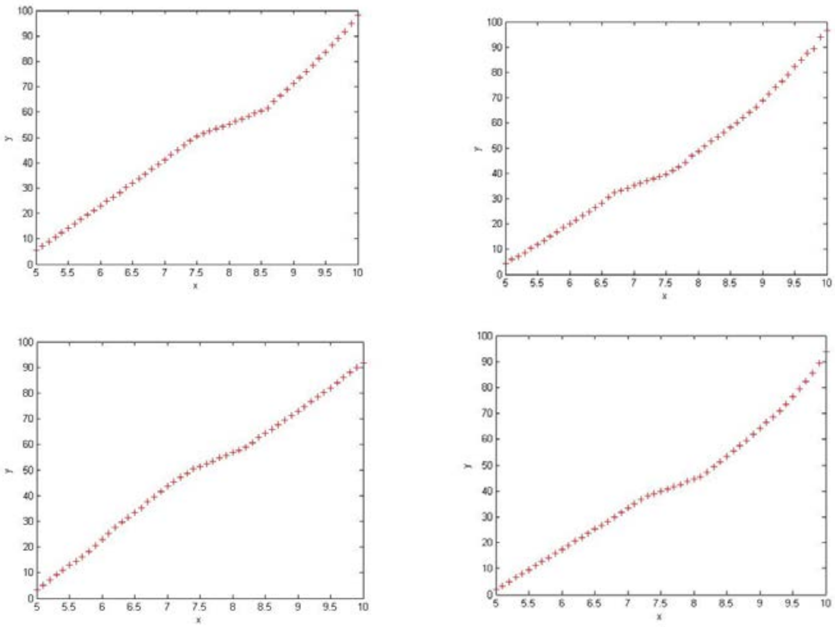


图 2.四个不同样本个体特性对比

4.不同拟合方法的比较

由于不同的集合方法在适应条件不同时各有优劣，所以我们尝试几种拟合方法来探索和优化拟合的方法。

4.1 多项式拟合方法（Polyfit 拟合）

给定数据点（X_i,Y_i）(i=0,1,2......,m)，利用 S（x）=ax^3+bx^2+cx+d，使得

$$I = \sum_{i=0}^{m}[p(x) - Y_i]^2 = \min \qquad \cdots\cdots\cdots\cdots\cdots\cdots (1)$$

求出（1）式中的唯一解，即可解出 S（x）中的各项的系数。

随机选出 250 组的八个点来运用上述公式进行多项式拟合，通过某一样本个体的定标成本函数：

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

求出每个个体的定标成本，再运用校准方案总体成本函数：

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

计算出平均成本。

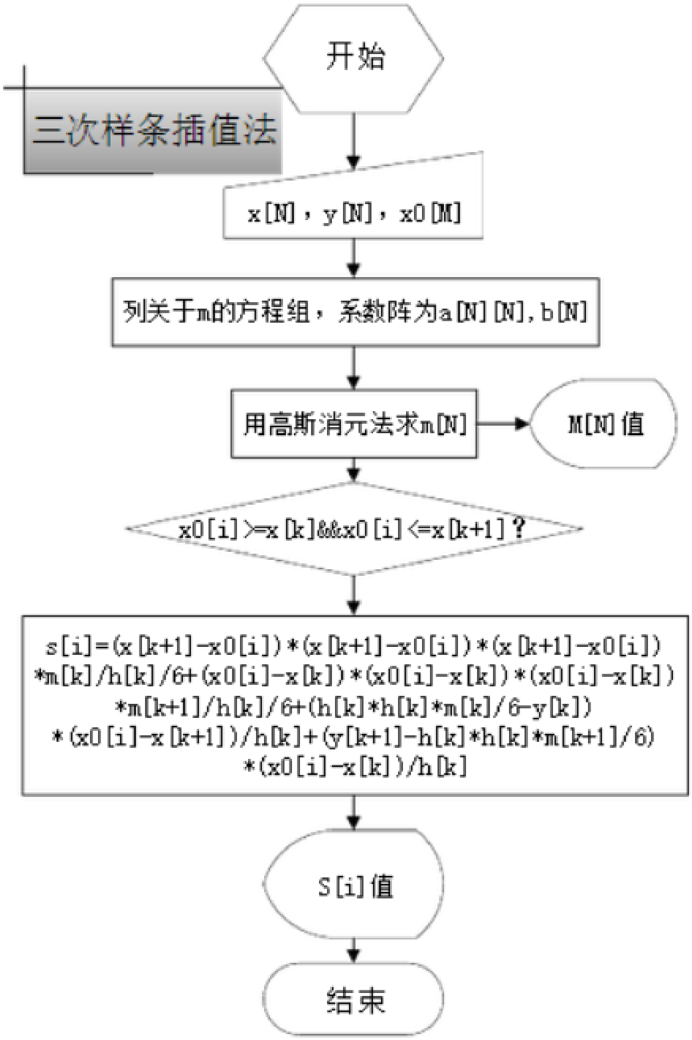
4.2 三次样条插值法

三次样条插值（简称 Spline 插值）是通过一系列形值点的一条光滑曲线，数学上通过求解三弯矩方程组得出曲线函数组的过程。

函数 S(x)∈C2[a,b] ，且在每个小区间[xj,xj+1]上是三次多项式，其中 a =x0 <x1<...< xn= b 是给定节点，则称 S(x)是节点 x0,x1,...xn 上的三次样条函数。若在节点 x j 上给定函数值 Yj= f (Xj).(j =0, 1, , n) ，并成立 S(xj) =yj .(j= 0, 1, , n) ，则称 S(x)为三次样条插值函数。

流程图如下：

图 3.三次样条差值法



由于本课题运用了 MATLAB 软件，所以调用其中的插值函数 spline，再输入相关的数据样本，就可以直接实现三次样条插值。

4.3　结论

通过对两种方法多组数据产生的平均成本进行对比和分析，可以计算得出利用三次样条插值法所得出的平均成本较低，所以我们决定运用三次样条插值法，来确定数据样本的拟合表达式

5 方法比较与选择

5.1.穷举法

穷举法是思维量较少，最为直观的方法，是最容易被想到的方法，这是穷举法最大的优点。

但是，在庞大繁杂的数据面前，穷举法的弱势便得以体现。大数据所带来的多次选点和多次拟合，非人力甚至计算机所能接受，而且耗时即为漫长，故本文放弃此方法。

5.2.遗传算法

遗传算法是模拟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。首先组成一组候选解，其次依据某些适应性条件测算这些候选解的适应度，再根据适应度保留某些候选解，放弃其他候选解，最后对保留的候选解进行操作，生成新的候选解。

然而，此方法虽然有效，而且准确率较高，但是编程过于复杂，对于解决此类问题性价比不高。故本文放弃此方法。

5.3.模拟退火法

模拟退火来自冶金学的专有名词退火。具体实现方法大致为：首先由一个产生函数从当前解产生一个位于解空间的新解；之后计算与新解所对应的目标函数差、判断新解是否被接受，如可接受，则在此基础上进行下一轮，如不可接受，则在上一轮基础上进行下一轮迭代。随着温度的降低（迭代次数的提高），可接受变异误差范围逐渐减小，最终趋于稳定，即得出最优解 。

此方法思路清晰明了，实现方便，对计算机运算能力，编程实现要求较为合理，实行的经济性有效性较前文提及 2 个方法均有显著优势。故，本文采用模拟退火法。

6 模拟退火法解决问题

6.1基本原理

本组选用模拟退火算法，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。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。用固体退火模拟组合优化问题，将内能E模拟为目标函数值f，温度T演化成控制参数t，即得到解组合优化问题的模拟退火算法：由初始解i和控制参数初值t开始，对当前解重复“产生新解→计算目标函数差→接受或舍弃”的迭代，并逐步衰减t值，算法终止时的当前解即为所得近似最优解，这是基于蒙特卡罗迭代求解法的一种启发式随机搜索过程。具体的求解过程可以参照流程图。

其算法流程如下：

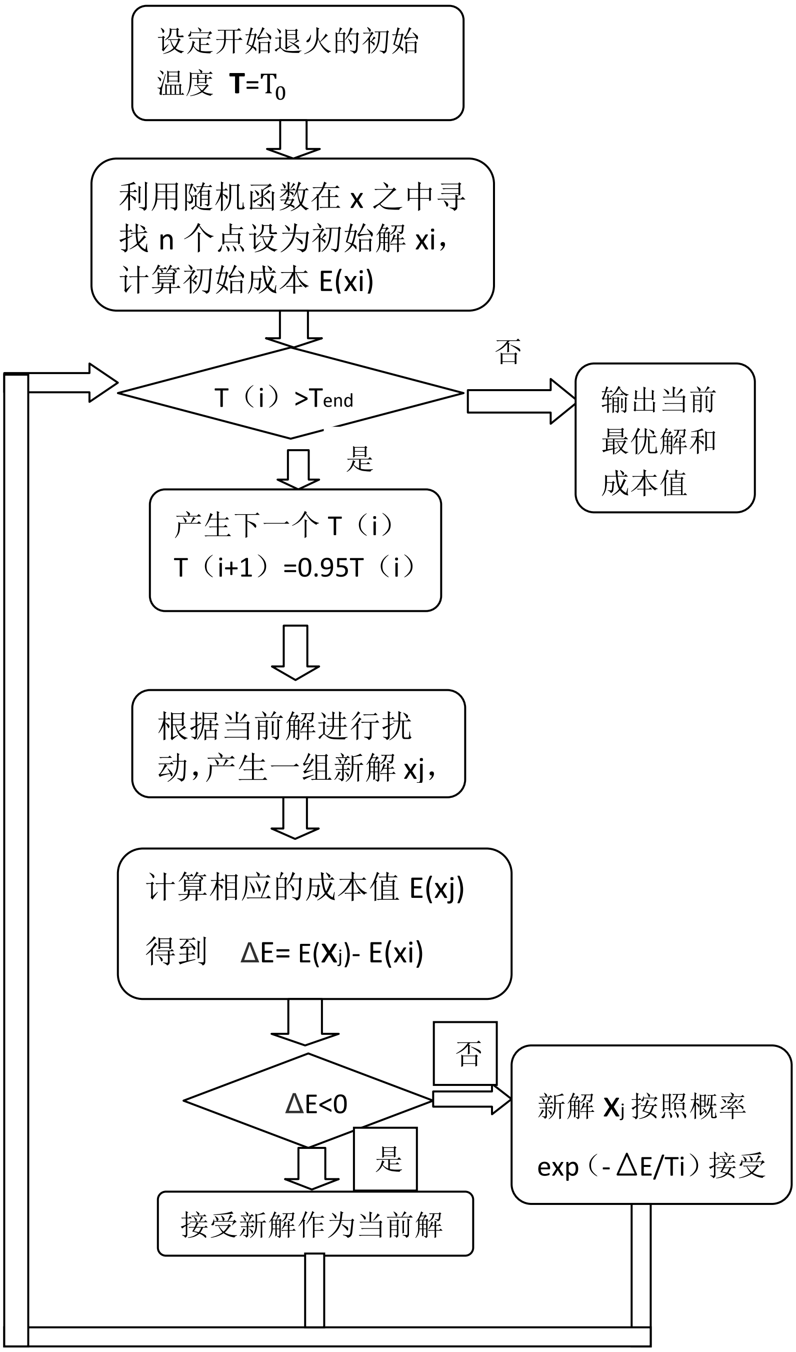


图 4.退火算法流程图

6.2 通过拟合确定成本

建立两个矩阵，Xij（i=1,2,…400,；j=1,2,…,51），Yij （i=1,2,…,400；j=1,2,…,51），某个解(s_1，s_2，…，s_7)

（1）通过拟合求出对应关系f

设对应关系为Y=f(X)。三次拟合法下，对每个Xi，只需对取得的七个点进行三次多项式拟合，即可得fi（X），不必让每一个数据全拟合，大大减少了拟合所花费的时间和空间。
三次样条插值法中，Xi，sk-1到Xi，sk之间的Y-X关系由过Sk-1，Sk，Sk+1，Sk+2四点的三次曲线确定，设曲线方程为$Y=a1X_3+a2X_2+a3X+a4$。由四点的值可以求得对应的系数值。所以对每个i，可以得到一个分段函数形式的Y=f (X)
（2）由拟合结果得到对应成本值
对于矩阵Xij（i=1,2,…,400; j=1,2,…,51）中的每一个值，代入Y=f(X)中，得到矩阵Y·ij（i=1,2,…,400; j=1,2,…,51）。
先对每个样本个体求出定标成本，然后再根据样本个体计算校准方案的总体成本

6.3通过模拟退火选择出最优解

设定温度T为控制参数，上文所得到的函数f而得到的成本作为内能，运用框图中的控制参数T的衰减函数来降低温度，使得成本（内能）也逐渐降低，直至趋于全局的最小值，再通过比较和分析所得结果的运算时间和成本，可信度等，总结出最优的解。

6.4算法的实现及程序运行结果

详细算法请见附录，其中代码选用模拟退火法和三次样条插值拟合，取点个数为7个，（由于在其他拟合法或取点个数的情况下，代码差别不大，因此仅附上得到最优解的代码。若要测试其他情况，只需修改元素个数和初值，或者修改拟合函数）
不同点数比较：

点数	选点方案	成本
5	【3,14,26,39,49】	129.7755
6	【3,12,22,31,42,50】	107.3040
7	【2,9,20,26,34,44,50】	95.2830
8	【2,9,20,26,34,42,44,50】	97.6323

最好的运行结果为三次样条插值拟合下，取【2,9,20,26,34,44,50】，成本为95.2830。

7.结果分析

通过模拟退火法和三次样条插值法得出的最低成本为95.2830。取点的个数本文借助了前人经验选定为7个，但是效果一般，多次迭代后成本依然有95之多。本文考虑了是否是由选点因素所造成的于是更改了选点个数，进行了取5、6、8个点的尝试，结果如上图所示。可以看出，前人的7个点的取法是较为科学的，其余取点方案相对于7点（尤其是5、6）点相差较大。因此舍弃其他取点方案，选择取出7个点的方案。

鉴于学期初小组内成员参加了数学建模竞赛，当时采用了最小二乘法拟合，2次拟合等多种方法，因此在此希望尝试改变拟合方式，试图得出更优解。然而，2次拟合法误差明显较大，均在110之上，最小二乘法拟合所得结果与三次样插条法差距不大，得出结果在96附近。然而最小二乘法程序编写更为复杂，经济性不高，故本文依旧推荐三次样插条法进行拟合。

8.参考文献

[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；
[2] 统计推断报告第 xx 组设计报告；

[3] 肖柳青、周石鹏。《随机模拟方法与应用》 上海交通大学出版社。
[4] 庞峰。《模拟退火算法的原理及算法在优化问题上的应用》。
[5] 黄世国、耿国华。《三次样条插值的逆扩散图像增强算法》

附录：

1.算法主程序

```
data=csvread('20150915dataform.csv');  %读入数据值
```

```
better_array=zeros(1,7);
x_t=zeros(1,7);
y_t=zeros(1,7);
temp=100;
sum_cost=0;
better_cost=0;
init_cost=250; %对成本进行初始化

while init_cost>=110  %对成本进行粗调，使成本在110之内，减少了之后退火算法的复杂度
    sum_cost=0;
    array0=([1,7]);
    array1=([8,14]);
    array2=([15,21]);
    array3=([22,28]);
    array4=([29,35]);
    array5=([36,42]);
    array6=([43,51]);
    sum_array=[array0,array1,array2,array3,array4,array5,array6];
    for i=1:400
        x_data=data((2*i-1),:);
        y_data=data((2*i),:);
        x_t=data((2*i-1),sum_array);
        y_t=data((2*i),sum_array);
        sum_cost=mycurvefitting0(y_data,x_t,y_t)+sum_cost; %计算当前总成本值

    end
    init_cost=sum_cost/400;
    better_cost=init_cost;
    better_array=sum_array;
end
[s1,s2]=output(init_cost,data,better_array,time); %s1为最优成本所选点
                                                  %s2为最优成本值
disp(s1);
disp(s2);
```

2.模拟退火程序

```
function [w,q] =output(init_cost,data,better_array,temp)

x_tq=zeros(1,7);
y_tq=zeros(1,7);
while temp >0.1      %运用退火的温度作为结束算法的依据
```

```
    disp(init_cost);
    temp=temp/(1.05);
    sum_cost=0;
    for i=1:7
        a=1;
        while a==1
            x_tp=better_array(i);
            c=randi(3);
            x_tp=x_tp+c-2;
            if (x_tp>0&&x_tp<=51)    %保证x在1到51的范围之内
                a=0;
            end
        end
        better_array(i)=x_tp;
    end
    clear i;
    for i=1:400
        x_data=data((2*i),:);
        y_data=data((2*i-1),:);
        x_tq=data((2*i-1),better_array);
        y_tq=data((2*i),better_array);
        sum_cost=mycurvefitting0(y_data,x_tq,y_tq)+sum_cost;  %计算当前总成本的值
    end
    init_cost=sum_cost/400;
    e=exp(-(init_cost-sum_cost)/time*100);
    if(init_cost<=better_cost)
        best_array=better_array;
        best_cost=init_cost;
        continue;
    end
    if(rand>e)      %, 根据粒子在温度temp时趋于平衡的概率不接受新产生的值
            better_array=best_array;
    end
    w=best_array;
    q=best_cost;
end
```

3.拟合与成本分析程序（test_ur_answer）

```
function f = mycurvefitting0(y_data,x_t,y_t)

for i=1:51
   xd(i)=5+(i-1)*0.1;             %统计x的所有值
end
y=interp1(x_t,y_t,xd,'spline');     %运用三次样条插值法拟合y值
```

```
fir_cost=0;
for j=1:51                    %计算单个个体的成本
    d_t=y(j)-y_data(j);
    d=abs(d_t);
        if d<=0.4
            fir_cost=fir_cost;
            continue;
        end
        if d<=0.6
            fir_cost=fir_cost+0.1;
            continue;
        end
        if d<=0.8
            fir_cost=fir_cost+0.7;
            continue;
        end
        if d<=1
            fir_cost=fir_cost+0.9;
            continue;
        end
        if d<=2
            fir_cost=fir_cost+1.5;
            continue;
        end
        if d<=3
            fir_cost=fir_cost+6;
            continue;
        end
        if d<=5
            fir_cost=fir_cost+12;
            continue;
        end
        if d>5
            fir_cost=fir_cost+25;
            continue;
        end
end
sm_cost=fir_cost+12*7;
f=sm_cost;
end
```

4. （**test_ur_answer**）

```
my_answer=[ 2,9,20,26,34,44,50 ];
my_answer_n=size(my_answer,2);

% ±ê×¼Ñù±¾ÔÊ¼Êý¾Ý¶ÁÈë
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;


% ¶¨±ê¼ÆËã
index_temp=logical(my_answer_gene);
x_optimal=x(:,index_temp);
y0_optimal=y0(:,index_temp);
for j=1:nsample
    % Çë°ÑÄãµÄ¶¨±ê¼ÆËã·½·¨Ð´Èëº¯Êýmycurvefitting
    y1(j,:)=mycurvefitting(x_optimal(j,:),y0_optimal(j,:));
end

% ³É±¾¼ÆËã
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;
```

```
% ÏÔÊ¾½á¹û
fprintf('\n¾¼ÆËã£¬ÄãµÄ´ð°¸¶ÔÓ¦µÄ×ÜÌå³É±¾Îª%5.2f\n',cost);
```