

参考他人报告或代码的声明

统计推断课程，2015 年秋季学期第 04 组，成员宁林, 学号 5140309417，王宝来, 学号 5140309350，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
算法描述	《统计推断在数模转换系统中的应用》，刘长风，2014 年秋季学期，组号 21 参考了该组对收敛过早问题的解决方案

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。

统计推断在数模转换系统中的应用

04 组：宁林 5140309417 王宝来 5140309350

摘要：使用 matlab 工具对某型电子产品的监测模块进行定标。参考现有课程资料，对给定的数据进行分析后选定适当的拟合以及取点方案的优化，以得到较低的定标总成本的方案。

关键词：Matlab、校准定标、拟合、启发式算法

1、引言

本课题来源于工业生产中常见的定标问题，进行数学建模后，即是对某一特定传感器个体，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。

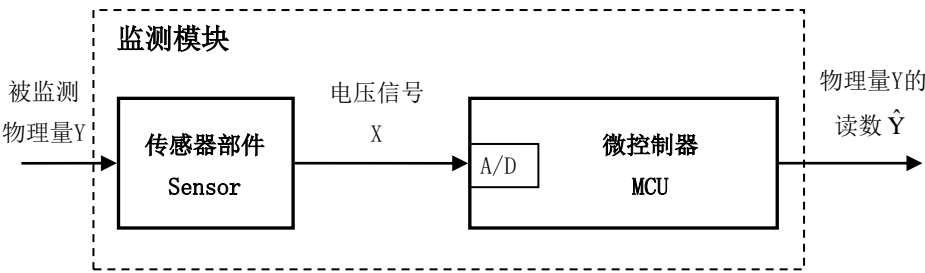


图 1 监测模块组成框图

2.、拟合方案

在选择拟合方案之前，我们对课题中的传感器部件进行了一定的了解，其输入输出特性大致如图所示

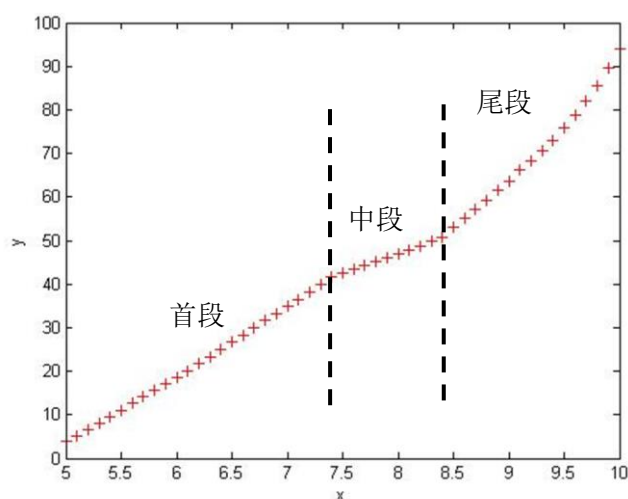


图2 传感特性图示

将其主要特征归纳如下：

- Y 取值随 X 取值的增大而单调递增；
- X 取值在[5.0,10.0]区间内，Y 取值在[0,100]区间内；
- 不同个体的特性曲线形态相似但两两相异；
- 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；
- 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；
- 不同个体的中段起点位置、终点位置有随机性差异。

由于我们不可能将所有点都选为特征点（这种方法会导致成本过高），因此必然会选择性的丢失部分数据。因此针对有限个特征点所选择的拟合方案就显得颇为重要。基于以上分析，我们初步选择了多项式插值、spline 插值和 pchip 插值三种插值方法进行比较。得到的结果如下表：

插值方法	取点个数	取点方案	定标成本	平均残差平方和
Spline 插值	6	1、11、22、31、43、51	97.8710	32.8125
Pchip 插值	5	5、16、27、38、48	88.9290	28.2559
3 次多项式插值	5	1、13、25、36、51	125.3742	73.5986
4 次多项式插值	6	1、10、25、36、44、51	111.8590	51.3200
5 次多项式插值	7	1、8、20、28、38、45、51	111.2400	29.9647

表2 各插值方法结果对比

表格中的定标成本依据以下函数确定：

1、单点定标成本：

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

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。

单点测定成本在本课题中指定为 12;

2、某一样本的定标成本由: $S_i = \sum_{j=1}^{51} s_{i,j} + q \cdot n_i$ 得到, 式中 n_i 表示对该样本个体定标过程中的单点测定次数。

3、校准方案总成本由 $C = \frac{1}{M} \sum_{i=1}^M S_i$ 得到。

从表格中看出, spline 插值、pchip 插值和 5 次多项式插值对传感器特性拟合的较好, 但由于 5 次多项式插值产生的特征点过多, 导致其定标成本过高, 不适合在本课题中使用。综合以上结果, 我们选择了 pchip 插值作为我们的拟合方法。

3、基于遗传算法选取特征点

好的拟合方法可以让结果更加精确, 而恰当的选点策略则是在保证精确性的前提下, 有效的降低我们定标所需的成本。

事实上, 特征点的选取是一个十分重要而困难的问题, 被称为 NP-Hard 问题。以课题所给的数据为例, 从 51 个点中选取一定数量的特征点, 这个选取方法的计算量已经不是计算机所能胜任的。因此, 我们根据课程讲座等所了解到的知识, 决定采用启发式算法来避免暴力穷举法所带来的低效率。

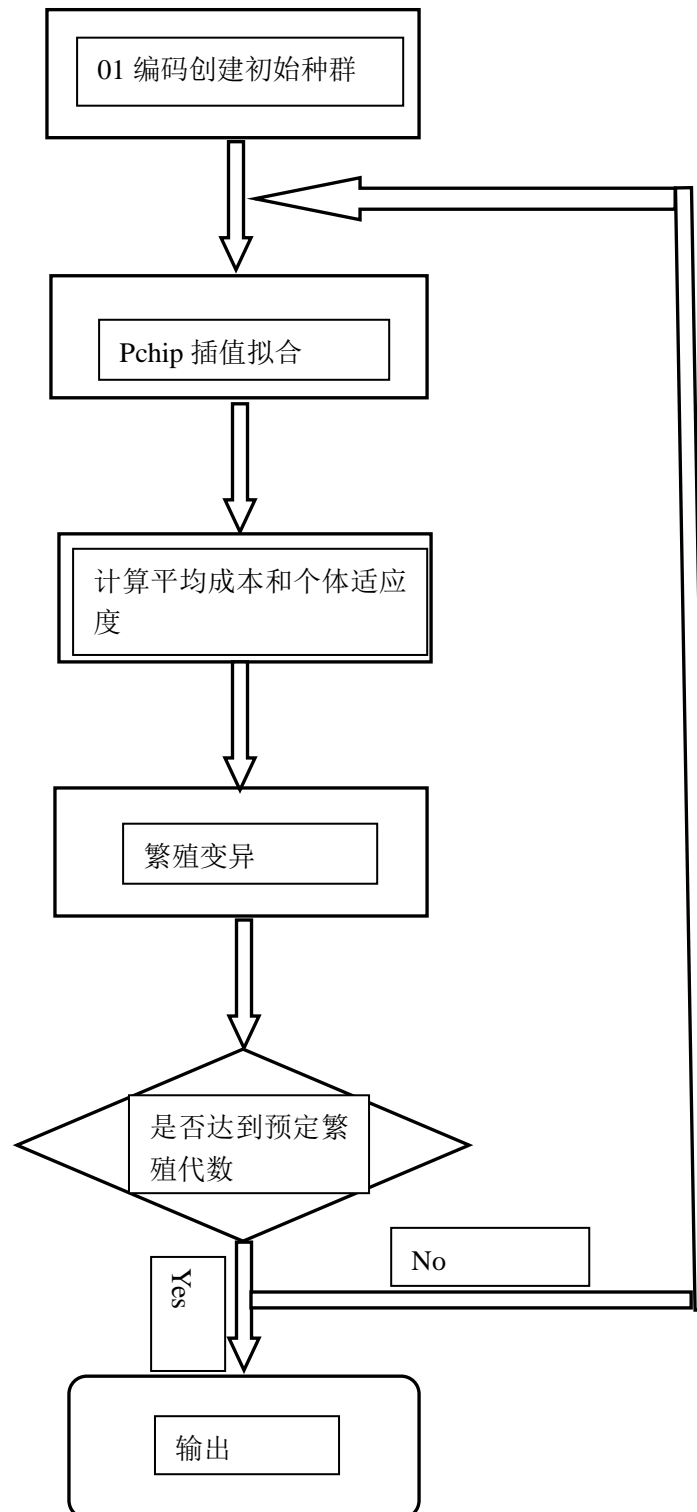
启发式算法中比较常见的有遗传算法、退火算法等。基于课程讲座和课程资料, 我们还是选择了较为常见的遗传算法作为求解问题的途径。

作为一个十分有效的启发式算法, 遗传算法具有以下特点:

- (1) 遗传算法从问题解的串集开始搜索, 而不是从单个解开始, 覆盖面大, 利于全局择优。
- (2) 具有内在的隐并性和很好的全局寻优能力, 可以同时处理群体中的多个个体, 即对搜索空间中的多个解进行评估, 减少了陷入局部最优解的风险, 同时使算法本身易于实现并行化。
- (3) 应用范围大, 采用概率化的寻优方法, 能自动获得和指导优化的搜索空间, 自适应地调整搜索方向, 不需要确定的规则。

(4) 直接对结构对象操作,不存在求导和函数连续性的限定,也不用搜索空间的知识及其他的辅助信息。仅用适应度函数来评价个体,在此基础上进行遗传操作。适应度函数不仅不受连续可微的约束,而且定义域可以任意设定。

(5) 遗传算法在适应度函数选择不当的情况下可能收敛于局部最优而不能达到全局最优。限于篇幅,在此就不列出遗传算法的具体过程,仅给出算法流程图:



4、总结与思考

在具体的算法实现过程中，我们也遇到了许多问题，以致无法得到令人满意的结果，甚至一度考虑重写算法。但经过我们的深入讨论，发现基于初稿的算法还有许多可以改进的地方。比如对循环的重写，过多的循环导致程序运行异常缓慢，这一点经过改进后达到了显著的效果。

此外，我们还发现，在前期的程序调试中，总是会出现过早收敛，即种群过早“成熟”的问题。在参考了往届同学的算法描述后，我们对变异的概率进行了适当的调整，并且采取了精英保留策略，使得程序总体的收敛速度放缓，有利于得到更优解。

综合以上的讨论，我们认为遗传算法在全局优化中表现良好，由于其采用的是启发式的搜索方法，极大的避免了暴力穷举带来的巨大时间成本，在该课题所设的场景下具有重要的现实意义。

5、参考文献

- [1] 袁焱.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] 上海交大电子工程系.统计推断在数模转换中的应用课程讲义[EB/OL].ftp://202.120.39.248.
- [3] 百度百科.遗传算法[J/OL].http://baike.baidu.com/link?url=DeeWvJb3arc-Bq1DhMgNzQ5-rJDxKLA9Yk032MY6jnFdRU0een6JVxpFln_pa3Ux_WmsZQLCu-yDiTBCNCTK

6、附录

6、1 词汇表

拟合

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。

插值（内插）

通常有两种含义。

- 一是指曲线必须通过若干已知离散数据点的一种拟合；
 - 二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。
- 本课程中，“插值”一词主要指上述第二种含义。

搜索算法

利用计算机的高性能运算能力，有目的地检验一个问题之解空间包含的部分或全部可能情况，从而求得问题的优化解的做法。

启发式搜索

在搜索问题解空间时，对当前已搜索的位置进行评估，寻找认为最好的下一步搜索方向，从这个（或这些）方向进行搜索直到目标。启发式搜索可以避免简单穷举式搜索效率低（甚至不可完成）的弊端。

课程自定义词汇：单点测定，单点测定成本

在本课题中，对 X 取某一定值时 Y 的真实取值进行测量确定的过程。完成单点测定要付出一定成本，称为单点测定成本。

课程自定义词汇：单点定标误差成本

在本课程中，系统完成定标后，对物理量 Y 的单点读数（即系统对 Y 的估测值）与该点 Y 实测值之间存在误差，按一定规则把该误差折算为一种成本。

6、2 符号表

X	传感部件的输出电压
Y	传感部件的输入，即被监测物理量
\hat{Y}	监测模块的输出，即监测模块将传感部件输出 X 所转换成的读数(对 Y 的估测值)
$y_{i,j}$	第 i 个样本之第 j 点对应的 Y 实测值（来自标准样本数据库）
$\hat{y}_{i,j}$	第 i 个样本之第 j 点对应的 Y 估测值
$s_{i,j}$	第 i 个样本之第 j 点对应的单点定标误差成本
q	单点测定成本
n_i	对样本 i 定标过程中的单点测定次数
S_i	对样本 i 的定标成本
M	标准样本数据库中的样本总数
C	基于标准样本数据库评价一个校准方案，算得的该方案总体成本

6、3 代码清单

BCD.m

```
function x = BCD(binary)
x = [];
for i = 1 : 51
    if(binary(i) == 1)
        x = [x i];
    end
end
end
```

Combine.m

```
function combine = combine(N,chrom,gap,minput,fitness)
combine = zeros(N,51);
randInt = RandInt(1:2,1);
point = RandInt(1:51,1);
n = 1;%生成的个体数目
if randInt == 1
    while n <= N
        if(rand(1,1) < gap)
            [row1,row2] = select(N,chrom,minput,fitness);%获得要进行交叉
            繁殖的两个个体
```

```

        combine(n,:) = [chrom(row1,1:point)
chrom(row2,point+1:51)];
        n = n + 1;
    end
end
else
    while n <= N
        if(rand(1,1) < gap)
            [row1,row2] = select(N,chrom,minput,fitness);%获得要进行交叉
繁殖的两个个体
            combine(n,:) = [chrom(row1,1:point)
chrom(row2,point+1:51)];
            n = n + 1;
        end
    end
end
end
end

```

creatPop.m

```

function chrom = creatPop(N,k)
chrom = zeros(N,51);
for i = 1 : N
    indi_selected = RandInt(2:50,k);
    chrom(i,indi_selected) = 1;
end
chrom(:, [1,51]) = 1;
end

```

Get_fitness.m

```

function fitness = get_fitness(N,chrom,minput)
fitness = zeros(N,1);%储存种群中每个个体的适应度
for i = 1:N
    x = BCD(chrom(i,:))
    fitness(i) = 1/test_ur_answer(x,minput);
end
end

```

Main.m

```

N = 50;%初始种群个体数目
gen_max = 81;%设定最大遗传代数
best_cost = zeros(1,gen_max);%存储每代最优解的成本
max_fitness = 0;%存储最大适应度
gen_num = 2;%计数器，用于计算进化代数
gap = 0.9;%交叉遗传的存活率

```

```

p = 0.3;%变异的概率
k = 4;%除去第1个和第51个点外选定的点的个数
mPop = 6;%需要的最小点数，2-三次样条插值，4-三次拟合，5-四次拟合，6-五次拟合
best_Pop = [];%最佳选点方案
stand_gen = 10;%设定最大容忍代数
cur_gen = 1;%当前最优解相同代数

best = 200;
%读入数据
minput=dlmread('20150915dataform.csv');
chrom = creatPop(N,k);
while gen_num <= gen_max
    fitness = get_fitness(N,chrom,minput);%得到每个个体的适应度
    chrom1 = [chrom fitness];%为提高速度而对chrom1初始化
    chrom1 = sortrows(chrom1,52);%按照适应度进行升序排序
    chrom = chrom1(:,1:51);%获取排序后的chrom
    fitness = sort(fitness);%将fitness与chrom中的个体在顺序上进行对应
    if best > 1/fitness(N,1)
        best = 1/fitness(N,1);
        best_Pop = BCD(chrom(N,:));%最佳取点方案为适应度最大的那一组
        cur_gen = 1;
    else
        cur_gen = cur_gen + 1;
    end

    chrom = save_elite(N,chrom,gap,p,minput,fitness,mPop);%进行交叉遗传、
    变异并保留上代精英
    if(cur_gen == stand_gen)
        chrom = creatPop(N,k);
    end
    gen_num = gen_num + 1;
end

```

Mycurvefitting.m

```
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,'spline');
```

```
f = polyfit(x_premea,y0_premea,5);%三次函数拟合，记得修改最小取点数
```

```
y1 = polyval(f,x);
```



```

end

randInt.m
function RandInt = RandInt(R, n)

N = length(R);
t = 0;    % 处理过的记录总数
m = 0;    % 已选得的记录数

RandInt = [];
while 1
    U = rand;
    if (N-t)*U < n-m
        m = m + 1;
        RandInt(m) = R(t+1);
        % 若已抽取到足够的记录, 则算法终止
        if m >= n, break, end
    end
    t = t + 1;
end
end

```

Residual.m

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 用 于 计 算 平 均 残 差 平 方
和%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function result = residual()
mininput=dlmread('20150915dataform.csv');
[M,N]=size(mininput);
nsample=M/2; npoint=N;
y1=zeros(nsample,npoint);
x=mininput(1:2:M,:);
y0=mininput(2:2:M,:);
my_answer_gene=zeros(1,npoint);
my_answer_gene([1    8    20    28    38    45    51])=1;

index_temp=logical(my_answer_gene);
x_optimal=x(:,index_temp);
y0_optimal=y0(:,index_temp);
for i = 1:nsample
    % 请把你的定标计算方法写入函数mycurvefitting
    y1(i,:)=mycurvefitting(x_optimal(i,:),y0_optimal(i,:));
end

```

```

y2 = (y1 - y0).^2;
result = sum(sum(y2))/nsample
end

```

Save_elite.m

```

function result = save_elite(N,chrom,gap,p,minput,fitness,mPop)
result = combine(N,chrom,gap,minput,fitness);
result = varia(N,result,p,mPop);
fitness1 = get_fitness(N,result,minput);
result1 = [result fitness1];
result1 = sortrows(result1,52);%将result按照适应度进行升序排序
result = result1(:,1:51);%删除result中的第52列
result([1,2],:) = chrom([N-1,N],1:51);%将result中适应度低的个体换为上一种群
中的精英
end

```

Select.m

```

function [row1,row2] = select(N,chrom,minput,fitness)
P = zeros(1,N);
Sum = 0;
fitness_sum = sum(fitness);%计算适应度的总和，方便后面用于轮盘赌
for i = 1 : N
    Sum = Sum + fitness(i,1);
    P(1,i) = Sum / fitness_sum;
end
x = rand(1,1);
row1 = 1;
while x > P(1,row1)
    row1 = row1 + 1;%其中row1为轮盘赌选中的第一个个体
end
x = rand(1,1);
row2 = 1;
while x > P(1,row2)
    row2 = row2 + 1;%row2为轮盘赌选中的第二个个体
end
if row1 == row2
    select(N,chrom,minput,fitness);%若两次选中同一个体，则重新进行轮盘赌
end
end

```

Varia.m

```

function varia = varia(N,chrom,p,mPop)
for i = 1:N
    if(rand(1,1) < p)

```

```

        randInt1 = RandInt(1:51,1);
        chrom(i,randInt1) = 1 - chrom(i,randInt1);
        while sum(chrom(i,:)) < mPop
            randInt1 = RandInt(2:50,1);
            chrom(i,randInt1) = 1 - chrom(i,randInt1);
        end
    end
end
end
end
varia = chrom;
end

```

Test_ur_answer.m

```

%%%%%% 答案检验程序 2015-11-04 %%%%%%%%%
function cost = test_ur_answer(my_answer,minput)
my_answer_n=size(my_answer,2);

[M,N]=size(minput);
nsample=M/2; npoint=N;
x=zeros(nsample,npoint);
y0=zeros(nsample,npoint);
y1=zeros(nsample,npoint);
x=minput(1:2:M,:);
y0=minput(2:2:M,:);
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 i = 1:nsample
    % 请把你的定标计算方法写入函数mycurvefitting
    y1(i,:)=mycurvefitting(x_optimal(i,:),y0_optimal(i,:));
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(nsampl e,1)*my_answer_n;
cost=sum(si)/nsampl e;

% 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);
end
```