统计推断在数模转换系统中的应用

--传感特性校准

组号：08　姓名:师昊 学号:5130309718　　姓名:刘乐天 学号:5130309712

摘要:

为了给监测某项与外部环境有关的物理量（如温度、压力、光强等）的监测模块中的传感器部件的输入输出特性进行校准，我们采用采样进行拟合曲线的方法，然而就该系统而言，在实际过程中，若对每个样品都进行所有 51 组数值进行完整测定，则既费时又高成本。若能减少需要观测的数值组数量，则可以提高功效。但是为了精确绘制出特性曲线，我们却又不得不增大采样点数。想解决这一问题，就必须提出一个优化采样的算法。本文分别讨论了区间穷举法、遗传算法等方法，希望最终能以更小的工作量取得更大的成果。

关键词:

遗传算法 拟合 优化采样 区间穷举法 最优解

Abstract：

In order to carry on one monitoring module’s sensing property’s calibration, which is monitoring the relationship between something and external environment (such as temperature, pressure, light intensity and so on). For the sake of calibrating the system fully, we find the section points out by the calculation of the arc string distance first, then characterize the whole system characteristics by using the 5 feature points found by using genetic algorithm, after that was the comparison which involves different interpolation(fitting). Meanwhile, we rectify some part of the genetic algorithm.

Key words:

Genetic algorithm, polynomial fitting, interval exhaustive, optimized sampling scan

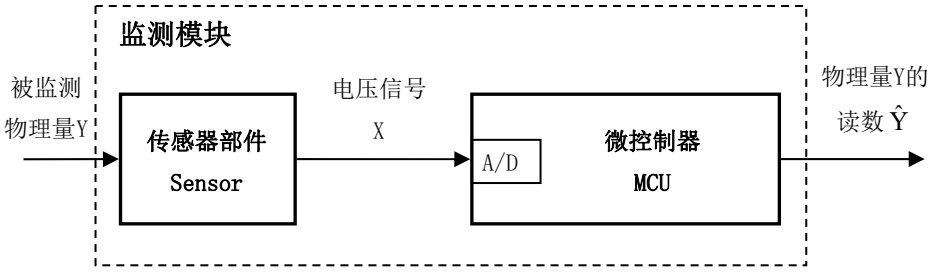
1. 引言

关于本次课题，为了给某型产品的一个检测模块寻求一个校准工序的最优方案，数据库提供的数据样本多达 469 个，而每个样本的测试点也都多达到了 51 个，如此庞大的一个数据容量，无疑会增加方案的成本。本课题的目的：1.选取出最优的特征点，减少成本；2.再将它们用不同的方案拟合在一起；3.比较之后，最终致力于寻找出成本最低的传感特性校准优化方案（包括选点方案和你和方式的选取）。

2. 问题来源与解决方案

2.1 数据对象来源：

从统计推断的任务测量对象中我们可以抽象出模型，如下图所示：



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）。

2.2 评价函数

【A】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。

【B】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。

【C】 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。

【D】 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。

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

2.3 选点方案的选取

2.3.1 穷举法

对于这个问题，如果要测量所有的测试点必定要耗费大量的成本，若是选取其中几个点，再取遍所有情况（以 7 点为例，就要有 115775100 种情况），很明显由于数据庞大，要比较全部的方案，成本太高，穷举法不可取

2.3.2 遗传算法

穷举法成本过高故不可取，所以我们采用老师课堂上提到的遗传算法来解决这个问题，下面是遗传算法的步骤：

A．初始化：设置进化代数计数器 t=0，设置最大进化代 20，随机生成 51 个条初始染色体

B．个体评价：对选择方式进行打分

C．选择运算:将轮盘选择算子作用于群体—概率=分数/总分数

D．交叉运算：根据交叉概率判断是否交叉

E．变异运算：根据变异概率判断每一位点是否变异（变异概率为 0.7）

F．修复运算：对不合法的子代进行修复（按照多减少补并且随机的原则）

G．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

2.3 拟合方式

2.3.1 多项式拟合

实验所提供的数据曲线如下图所示绝大部分都很接近一个多项式曲线。根据初稿的对实验数据的拟合，多项式拟合出的多项式曲线更具有拟合的准确性。在本实验中采用最高次项为三次的多项式拟合。

2.4.2 三次样条插值函数法

假设在[a,b]上测量有 n-1 个点，在[a,b]上划分。△:$a=x_0<x_1<\ldots<x_{n-1}<x_n=b$，则三次样条

插值法所得的函数 S(x)具有如下特征[1]：

【1】在每个小区间$[x_j,x_{j+1}]$上是三次多项式。

【2】在每个节点 $S(x_j)=y_j$。

【3】S(x)在[a，b]二阶导数连续。

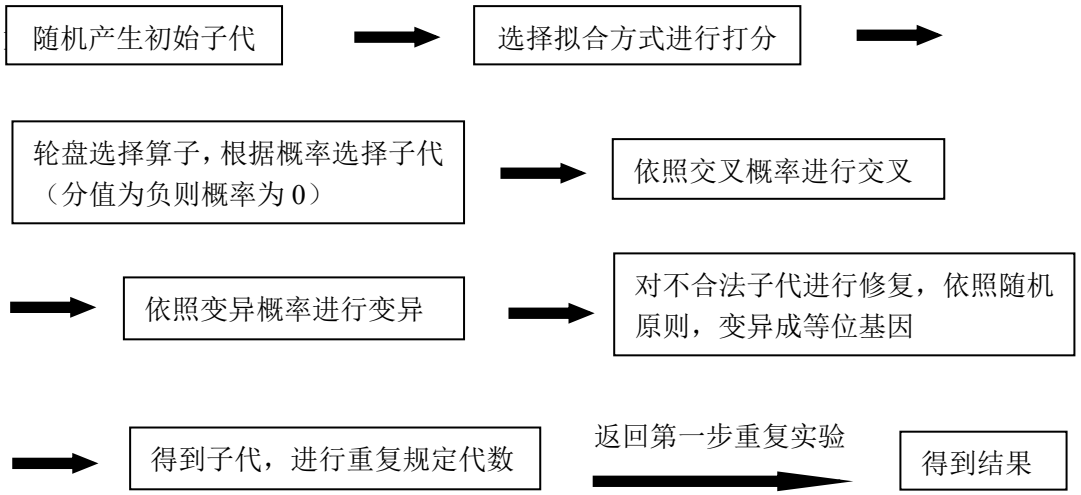
因为 S(x)在[a，b]上二阶导数连续，所以在每个节点，由连续性得 S(xj-0)=S(xj+0)；S’(xj-0)=S’(xj+0)；S’’(xj-0)=S’’(xj+0)。共 3n-3 个条件；除此之外，由插值可得 n+1 个条件，仍需两个，这两个条件有边值得出：

【1】给定断点 x0,xn 处的一阶导数值

【2】给定断点 x0,xn 处的二阶导数值（特别的 S’’(x0+0)=0, S’’(xn+0)=0 称为自然边值条件）

【3】周期性便捷条件：即以 b-a 为周期，S’’(x0+0)= S’’(xn+0)，S’(x0+0) =S’(xn+0)。这样求出的三次曲线比 2.4.1 中的多项式拟合应该要精确许多。

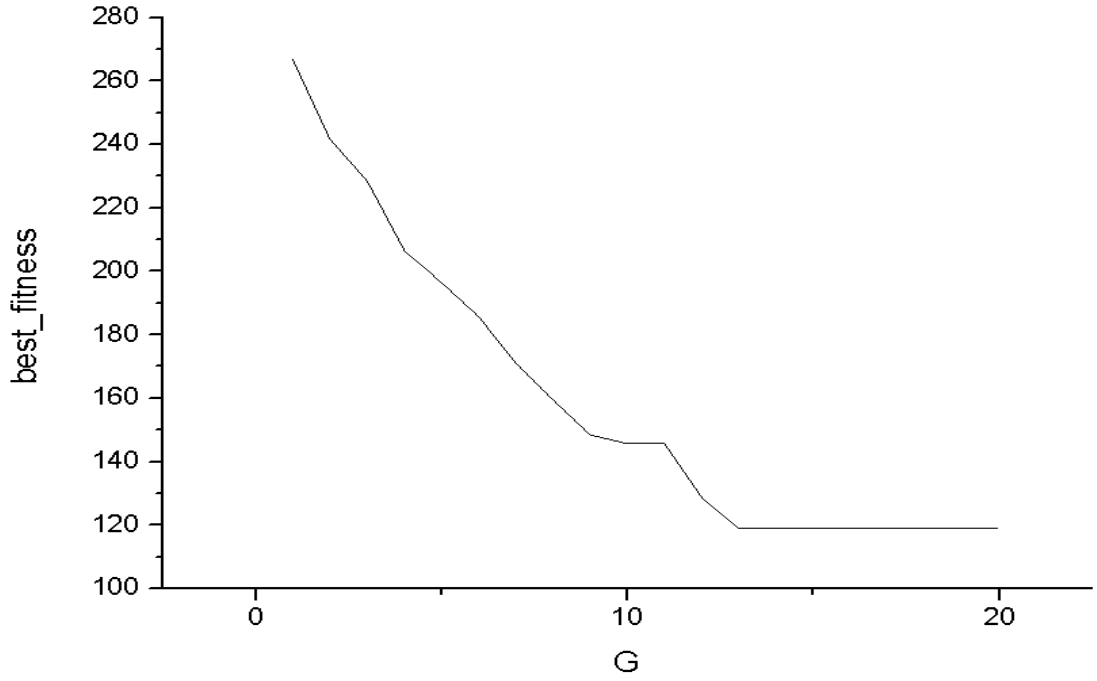
2.5 遗传算法的实现



我们采用了一个有 100 个种群，每个种群的染色体为 51 条，子代个数为 20 代，交叉概率为 1.0，变异概率为 0.7 的遗传算法。因为之前我们是采用随机从 51 个点选取出来几个点再进行打分、比较、选择、交叉、变异，而且 51*51<<100^20，所以我觉得我们的方法几乎已经遍历了所有的可能选点方案，所以的出来的结果是符合要求的。

3. 结果分析

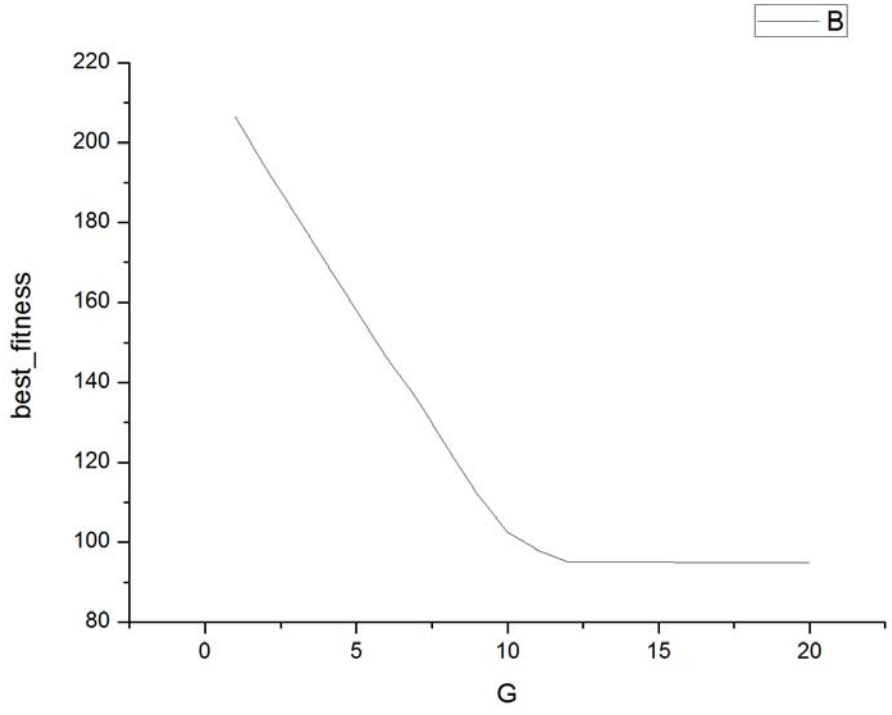
3.1 多项式拟合



最优成本收敛曲线

G 为代数 best_fitness 为最优成本
经 Matlab 运行程序及 Origin 作图，可得
最佳取点方案 q=2，14，24，34，50
取样成本 0119.3

3.2 三次样条插值



最优成本收敛曲线

G 为代数，best_fitness 为每一代的最优成本
经 Matlab 运行程序及 Origin 作图，可得
最佳取点方案 q = 1，9，22，28，39，50
取样成本 95.36

3.3 分析与讨论

【1】由上述两组遗传算法采用不同的拟合方式得出的最有成本的不同，我们可以根据分值看出，三次样条插值打分有效性更好好，多项式拟合（即三次拟合）有效性比较起来明显成本更高。
【2】95.36 的成本较之已经得到了很大的改善，说明三次样条插值法的方案是值得肯定的，所以最终我们采取了遗传算法的三次样条插值的方案，三次样条这种方案是更合理的。
【3】整体方法特征：
遗传算法本身就是模拟生物进化得到的结果。而在生物实验中，我们对区间庞大的连续量无法逐一去处理时，我们经常的处理方法就有预实验，得到大致的分布，再选取更加细致的区间进行进一步实验，得到最优结果。

4. 结论

【1】经过比较，三次样条插值法的最有成本效果优于三次多项式拟合法，因此我们得到 7 点数据后可采用三次样条插值估计整条曲线。
【2】保留最大值的遗传算法能加快遗传算法的收敛性，同时保证最优解不重新打破。对于区间比较均匀的 NP 问题，往往划分区间具有有效性，能加快收敛速度，得到最优解。
【3】模拟生物的多种算法都可运用于工程实践中来。

5. 收获

【1】初步了解了一种新的程序语言—matlab
【2】了解了如何用程序语言来解决实际问题
【3】大致了解了统计推断在实际生活中的应用
【4】初步了解了遗传算法的应用
【5】明白了学习生活中合作的重要性
【6】最后也要感谢老师的悉心讲解和解惑

参考文献：

【1】《遗传算法和组合优化》
【2】《概率论与数理统计》-上海交通大学数学系组编
【3】FTP 统计推断课件
【4】《matlab 教程》
【5】《DS 证据理论在雷达体制识别中的应用》（王 勇，毕大平　文献标识码：A 文章编号：CN51—1418(2oo5)o6—0003—04）

5. 结论

【1】经过比较，三次样条插值法的最有成本效果优于三次多项式拟合法，因此我们得到 7 点数据后可采用三次样条插值估计整条曲线。

【2】保留最大值的遗传算法能加快遗传算法的收敛性，同时保证最优解不重新打破。对于区间比较均匀的 NP 问题，往往划分区间具有有效性，能加快收敛速度，得到最优解。

【3】模拟生物的多种算法都可运用于工程实践中来。

5. 收获

【1】初步了解了一种新的程序语言—matlab
【2】了解了如何用程序语言来解决实际问题
【3】大致了解了统计推断在实际生活中的应用
【4】初步了解了遗传算法的应用
【5】明白了学习生活中合作的重要性
【6】最后也要感谢老师的悉心讲解和解惑

参考文献：

【1】《遗传算法和组合优化》
【2】《概率论与数理统计》-上海交通大学数学系组编
【3】FTP 统计推断课件
【4】《matlab 教程》
【5】《DS 证据理论在雷达体制识别中的应用》（王 勇，毕大平　文献标识码：A 文章编号：CN51—1418(2oo5)o6—0003—04）

4. 程序附录（选取程序中主要部分）

Main 函数：

```
%遗传算法主函数

 %pop_size: 输入种群大小

 %chromo_size: 输入染色体长度   %generation_size: 输入迭代次

数   %cross_rate: 输入交叉概率

 %mutat_rate: 输入变异概率
function [m,n,p,q] = GeneticAlgorithm(pop_size, chromo_size,
generation_size, cross_rate, mutate_rate, elitism)


global G ; %当前代

global best_cost;%最优成本

global best_fitness;%历代最佳适应值

global fitness_avg;%历代平均适应值矩阵

global best_individual;%历代最佳个体

global best_generation;%最佳个体出现代

global pop;%50 个种群的 51 个基因

global fitness_value;%每个个体分别成本
global data;

fitness_avg = zeros(20,1);
pop_size=100;
chromo_size=51;
elitism=true;
cross_rate=1;
mutate_rate=0.5;
```

```
best_individual=zeros(1,51);
data=csvread('20141010dataform.csv',0,0);
x=zeros(469,51);
y=zeros(469,51);

for i=1:469;

x(1,:)=5.0:0.1:10; %x 的值     y(1,:)=data(2*i,:); %y 值

end

clear i;
fitness_value(100)=0;
best_generation = 0;
pop=zeros(100,51);

initilize(100, 51);%初始化

best_individual=zeros(1,51);
best_fitness=-100000;

for G=1:20

    fitness(pop_size, chromo_size);  %计算适应度

    rank(pop_size, chromo_size);   %对个体按适应度大小进行排序

    selection(pop_size, chromo_size);%选择操作

    crossover(pop_size, chromo_size, cross_rate);%交叉操作

    mutation(pop_size, chromo_size, mutate_rate);%变异操作


n = -best_fitness;%获得每一次的适应度


disp "代数："G

disp "最优适应度"n

end

disp "最优适应度"-best_fitness
```

```
m = best_individual;%获得最佳个体

disp "最佳选点方案" q=decode(m)


clear;
```

成本函数 chengben.m

```
function [ s ] = chengben(x,y,A,B)
s=0;

if abs(A(1,y)-B(2*x,y))<=0.5
    s=0;
elseif abs(A(1,y)-B(2*x,y))<=1
    s=0.5;
elseif abs(A(1,y)-B(2*x,y))<=2
    s=1.5;
elseif abs(A(1,y)-B(2*x,y))<=3
    s=6;
elseif abs(A(1,y)-B(2*x,y))<=5
    s=12;

elseif abs(A(1,y)-B(2*x,y))>5
    s=25;
end

end
```

拟合计算：fitness.m

```
%计算种群个体适应度，对不同的优化目标   %pop_size: 种群大小

%chromo_size: 染色体长度


function fitness(pop_size, chromo_size)
global fitness_value;
global pop;
%global pop;   %global G;
global data;

for i=1:100

    xuanze(1,:)=decode(pop(i,:));%插值点
```

```
    xuanze_n=size(xuanze,2);
    for j=1:469
        x=data(1,:);
        y=data(2*j,:);
        x0=[5.0:0.1:10.0];

        fitx(1,:)=x(xuanze);
        fity(1,:)=y(xuanze);
        y0(1,:)=interp1(fitx(1,:),fity(1,:),x0,'spline' );
%进行三次线性插

值  %y1=polyfit(fitx(1,:),fity(1,:),3);y0(1,:)=polyval(y1,

x0);%多项式拟

合       %y0(1,:)=imresize(fitx(1,:),fity(1,:),x0,'bicubic

' );
      for k=1:51
fitness_value(i)=fitness_value(i)+chengben(j,k,y0,data);
      end
      clear y0;
    end

fitness_value(i)=fitness_value(i)+length(xuanze)*12*469;
fitness_value(i)=-fitness_value(i)/469;


    clear fitx;    clear fity;    clear xuanze;
end
```

选择：selection.m

```
%轮盘赌选择操作   %pop_size: 种群大小   %chromo_size: 染色体长度

%elitism: 是否精英选择


function selection(pop_size, chromo_size)
global pop;
global fitness_table;

for i=1:pop_size

    r = rand * fitness_table(pop_size);%随机生成一个随机数，在
```

```
0 和总适应度之间，fitness_table(pop_size)为总适应度*/
    first = 1;
    last = pop_size;
    mid = round((last+first)/2);
    idx = -1;
    while (first <= last) && (idx == -1)
        if r > fitness_table(mid)
            first = mid;
        elseif r < fitness_table(mid)
            last = mid;
        else
            idx = mid;
            break;
        end
        mid = round((last+first)/2);
        if (last - first) == 1
            idx = last;
            break;
        end
   end

   for j=1:chromo_size
       pop_new(i,j)=pop(idx,j);
   end
end

    p = pop_size-1;

for i=1:p
   for j=1:chromo_size

       pop(i,j) = pop_new(i,j);%若是精英选择，则只将 pop_new 前

pop_size-1 个个体赋给 pop，最后一个为前代最优个体保留*/

   end
end

clear i;   clear j;   clear pop_new;   clear first;
clear last;   clear idx;   clear mid;
```

变异：mutation.m

```
%单点变异操作     %pop_size: 种群大小
```

```
%chromo_size: 染色体长度   %cross_rate: 变异概率
function mutation(pop_size, chromo_size, mutate_rate)
global pop;

for i=1:pop_size
    if rand < mutate_rate
        mutate_pos = round(rand*51);
        if mutate_pos == 0
            continue;
        end
        pop(i,mutate_pos) = 1 - pop(i, mutate_pos);
    end
end

clear i;
clear mutate_pos;
```