统计推断在数模转换系统中的应用

第五组 李韬 学号：514030918 李伟豪 学号：5140309018

摘要 ：

本文主要阐述了在实际工程数据点数量多，随机性强的特点下，如何利用有限的测试点离散的情况下去模拟连续完整的变量，并使误差以及测量的成本尽可能的小。本文的标题为统计推断在模数数模转换系统中的应用，实际上是为某产品的一个检测模块批量生产设计一种成本尽可能低的传感器校准（定标工序）方案。

关键词：多项式拟合，三次样条插值，多项式插值，启发式搜索，遗传算法。

Application of Statistical Inference in DA Inverting System

Group 05 Li Tao 5140309183, Li WeiHao 5140309018

Abstract: This paper describes how to use the limited test point to implement the simulation of continuous variables, and make the cost as low as possible, under the circumstances that the actual engineering has a large amount of random data. This paper titled “Application of Statistical Inference in DA Inverting System”, it actually introduce how to design a sensor calibration program for a detection module of one product in mass production with costs as low as possible.

Key words: Polynomial fitting, Three times spline interpolation method, Polynomial interpolation, heuristicsearch, GeneticAlgorithm

1. 引言

当今世界早已进入了大数据的时代，因此大批量的处理数据已经成为了工程中不可或缺的一步。文章的标题是统计推断在模数数模转化中的应用，实际上是对于某产品一个检测模块批量生产设计一种低成本的方案。本文就该产品某模块的输入输出特性进行测量和定标方案设计进行初步探索。在实验中，因为数据量过于巨大，我们用到了启发式搜索算法，具体为遗传算法，选出对我们定标最有利的散点，通过合适的拟合方法实现对产品的定标。

2.数值处理方法：

2.1 多项式插值

多项式拟合函数（即 polyfit 函数）是 matlab 中用于进行曲线拟合的一个函数。其数学基础是最小二乘法曲线拟合原理。曲线拟合：已知离散点上的数据集，即已知在点集上的函数值，构造一个解析函数（其图形为一曲线）使在原离散点上尽可能接近给定的值。

调用方法：polyfit(x,y,n)。用多项式求过已知点的表达式，其中 x 为源数据点对应的横坐标，可为行向量、矩阵，y 为源数据点对应的纵坐标，可为行向量、矩阵，n 为你要拟合的阶数，一阶直线拟合，二阶抛物线拟合，并非阶次越高越好，根据拟合情况而定。

多项式在 x 处的值 y 可用下面程序计算。

y=polyval(a,x,m)

2.2 三次样条差值

早期工程师制图时，把富有弹性的细长木条（所谓样条）用压铁固定在样点上，在其他地方让它自由弯曲，然后沿木条画下曲线。成为样条曲线。

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

三次样条插值的特点是处处有二阶导数，故拟合的是一条光滑曲线。

这里主要使用的函数如下：

y=interp1(x,y,x0,'spline');

2.3 三次多项式插值(即 Hermite 插值)

Hermite 插值是利用未知函数 f(x)在插值节点上的函数值及导数值来构造插值多项式的，其提法为：给定 n+1 个互异的节点x_0，x_1，……，x_n上的函数值和导数值求一个 2n+1 次多项式 H2n+1(x)满足插值条件

$$H_{2n+1}(x_k) = y_k$$

$$H'_{2n+1}(x_k) = y'_k \quad k=0,1,2，……，n$$

如上求出的$H_{2n+1}(x_k)$称为 2n+1 次 Hermite 插值函数，它与被插函数一般有更好的密合度.

其基本思想是利用 Lagrange 插值函数的构造方法，先设定函数形式，再利用插值条件求出插值函数

2.4 三次样条差值与三次多项式插值的区别

pchip 指分段三次 Hermite 插值，调用形式：pchip(x,y,x_x)　x_x为插值点

spline 指三次样条插值,调用形式：spline(x,y,x_x)　x_x为插值点

spline 提供的函数 s(x)的构建方法和 pchip 里面的函数 p(x)完全相同，只不过在 X(j)处的斜率的选择方法不一样，spline 函数的 s(x)在 X(j)的二阶导数$D''_s(x)$也是连续的，这导致了如下结果： spline 更加光滑，也就是说，$D''_s(x)$是连续的。如果

数据是一个光滑函数的值，则 spline 更加精确。　如果数据不是光滑的，则 PCHIP 不会超过目标值，也不太震荡。pchip 建立的难度较小。

spline 比 pchip 光滑，样条的两阶导数连续，而 pchip 一阶导数连续。不连续的两阶导数隐含着不连续的曲率。人的眼睛可以检测出图形上曲率的不连续。另一方面，pchip 是保形状的，而 spline 不一定保形状。

对比如下：

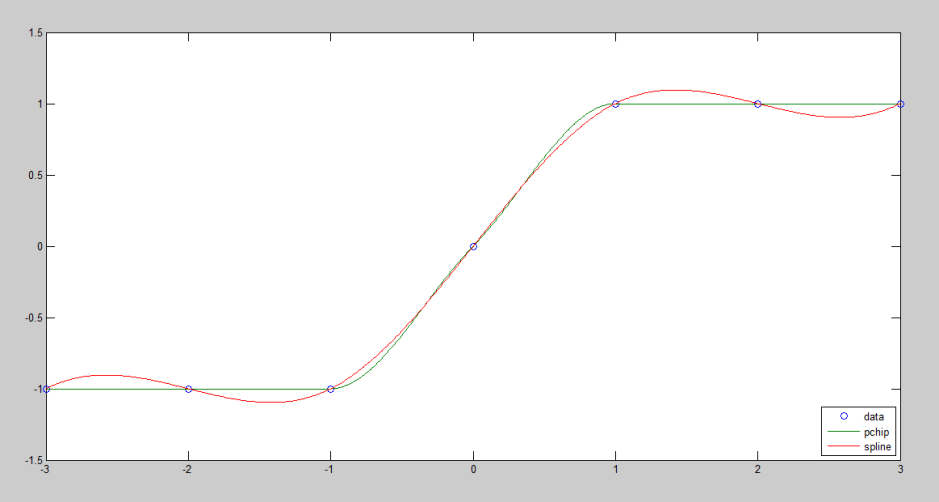


图 1

红色曲线--spline 插值 1

绿色曲线--pchip 插值 1

3. 遗传算法基本理论

3.1 简介

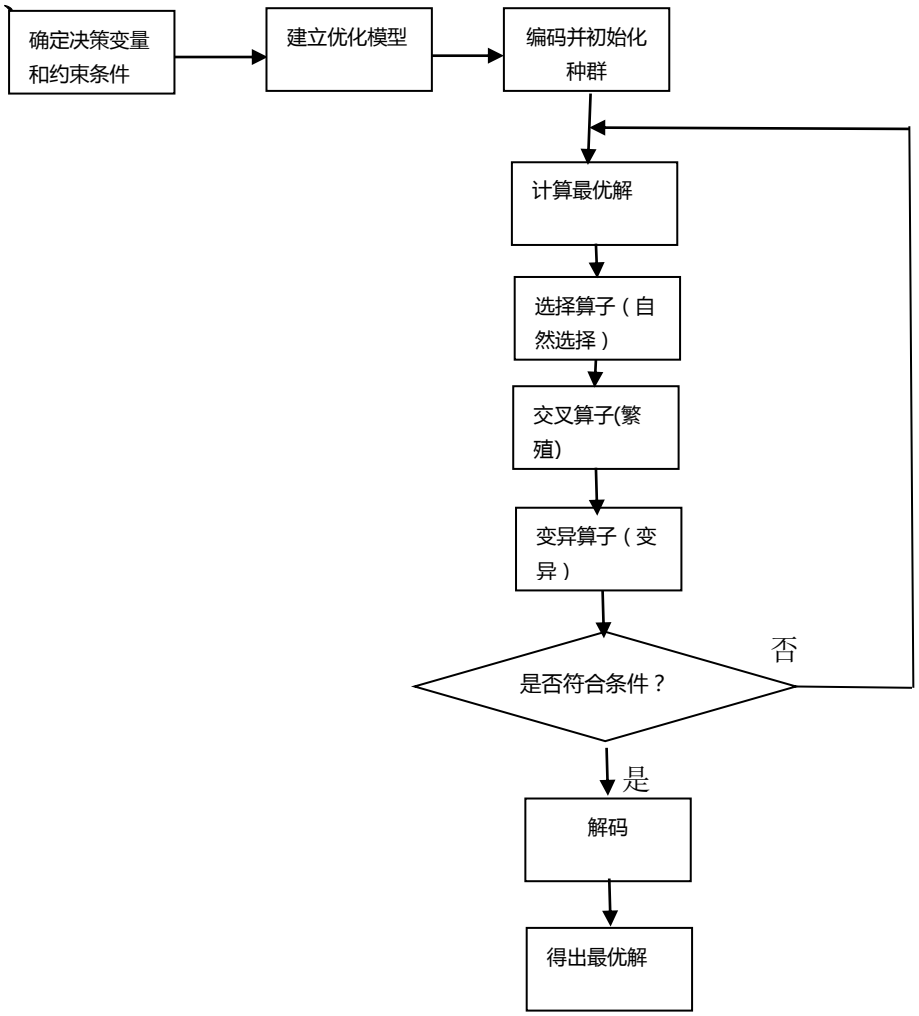
遗传算法基于自然界中生物遗传进化机理的模仿，针对不同的问题，很多学者设计了许多不同的编码方法表示问题的可行解，开发出许多不同的遗传算子来模仿不同环境下生物遗传特性。这样，不同的遗传算子构成各种不同的遗传算法。所有的遗传算法都有共同的特点，即通过对生物遗传和进化过程中的自然选择、交叉、变异机理的模仿，来完成对问题最优解的自适应搜索过程。基本遗传算法只使用选择算子（自然选择）、交叉算子（繁殖）、变异算子这三种基本遗传算子，其操作过程简单、容易理解，同时也是遗传算法的基本框架。

对于该课题，我们只采用了最基本的遗传算法寻找到问题的近似最优解。

3.2 个体适应度评价

在遗传算法中，个体适应度大小确定了该个体被遗传到下一代群体中的概率。个体适应度越大越容易被遗传到下一代，反之遗传到下一代的概率越小。个体的适应度根据实际解对具体问题的最优解的接近情况评价。

3.3 遗传算法基本流程



3.3.1 选择算子

模拟自然界中自然选择，优胜劣汰的过程。该算子接受某个体的适应度并对应出该个体的生存概率。并最终生成随机事件，决定该个体是否被淘汰。值得注意的是，适应度较高的个体（对应较优解），并不一定会存活。同样，适应度较低的个体也不是一定会淘汰。这给算法带来了更大的随机性，有利于寻找到更优质的解。

3.3.2 交叉算子

模拟自然界中自由交配过程。在自然选择中存活下来的个体，通过交配交换各自的基因。有可能产生更加优秀的个体。

3.3.3 变异算子

模拟自然界中的基因变异过程。变异是增加个体多样性的重要途径。引入变异算子可以增加解的多样性。但是变异率是一个重要的参数，应根据实际问题做具体调整。

4. 传感器定标方案设计

4.1 输入输出函数和采样点函数的设定

随机抽取给定数据中的若干组，画出 Y-X 图像。初步观察到，Y-X 图像基本符合三次函数的关系。一个三次函数的确定至少需要 4 个采样点。从理论上来说，采样点选取的数量越多，所得到的函数关系越精确。但是考虑到实际成本，我们最多取 9 个采样点，这样，程序会将采样点从 4 到 9 枚举，选取最优解作为最终结果。

4.2 确定最优取点方案

如果选取 9 个采样点，总共有C_{51}^{9}种方案。从时间复杂度来看，不可能通过暴力穷举获得该问题的最优解。所以我们采用遗传算法，得到问题近似最优解。遗传算法具有良好的启发性和自适应性，能极大减小算法的复杂度，得到结果非常逼近最优解。

4.2.1 适应度评价

个体（取点方案）的适应度和其测量成本有关。可以直观地看到，个体的测量成本越高，适应度越低。

(1) 单点定标误差成本

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

据本课程要求，单点成本由上式确定，其中$\hat{y}_{i,j}$是第 i 个样本，第 j 点的估测值。$y_{i,j}$表示第 i 个样本，第 j 点的实测值。从上式可以看出，我们对误差较大的点采取惩罚性措施，误差越大对应的成本越大。

(2)样本定标成本

$$S_i = \sum_{j=1}^{51} s_{i,j} + 12N_i$$

式中，N_i是选择的采样点个数。从表达式中，可以看到，样本成本不仅和测量误差有关，而且和选取采样点的个数有关（根据课程要求，每一个采样点需要 12 的固有成本）。这也就意味着，不能无限制地增加采样点。

(3)定标方案总成本

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

M 是样本总数。取所有样本成本的平均值即定标方案的总成本。

(4)成本与适应度对应函数

成本与适应度的关系，其实就是成本与个体存活概率的关系。为了选取出质量更高的个体，同时也为了防止种群“早熟”，存活概率必须选择得非常恰当。要想从理论上得到最优的对应方案，必须运用统计学方法，根据实际数据的分布情况来决定。我们选取平方关系对应存活概率，具体公式如下：

$$p=（C-C_{\min}）\hat{}2/C_{\max}C_{\max}$$

上式中，C 是某一选点方案的成本。$C_{\max}$和$C_{\max}$是种群中成本最大值和最小值。P 是存活概率。

之所以采用这样的选点方案，是因为这样选点能给高成本的个体足够的选择压，同时又能给低成本的个体足够的存活概率，将遗传漂变控制在适当的范围内。值得注意的是，在这种对应方案下，成本最小的个体，存活概率为 1，也就是说，它一定会遗传到下一代。同时，成本最高的个体，死亡概率为 1。

4.2.2 遗传算法参数设置

选用初始种群大小 100，遗传代数 10，变异率 0.3。交配父本母本对应基因发生交叉互换的概率为 0.5。

4.3 实验结果分析

4.3.1 采样点个数

采样点个数	4	5	6	7	8	9
最优解成本	99.5455	84.915	87.2135	93.1363	101.7453	111.8802

表 1

可以看到，采样点取 5 个或者 6 个可以获得相对成本小的方案。值得注意的是，取 7 个点可以得到质量比较高的解，同时由误差产生的成本相对较小。也就是说，在成本允许的情况下，取 7 个点也可能是一个兼顾成本和精度的方案。

4.3.2 遗传代数和最优解

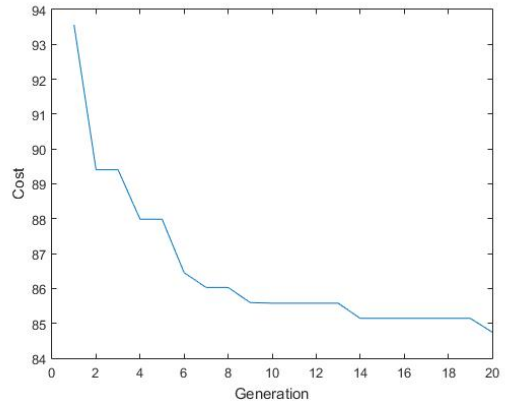


图 2 5 采样点下代数和最小成本关系

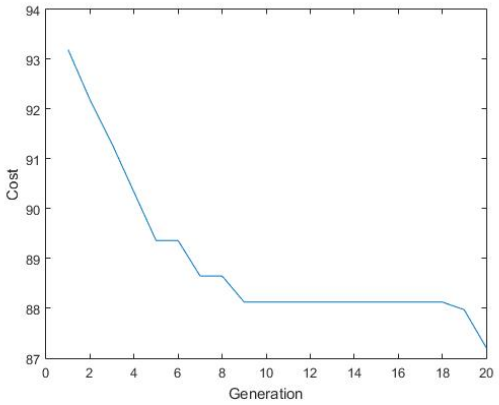


图 3 6 采样点下代数和最小成本关系

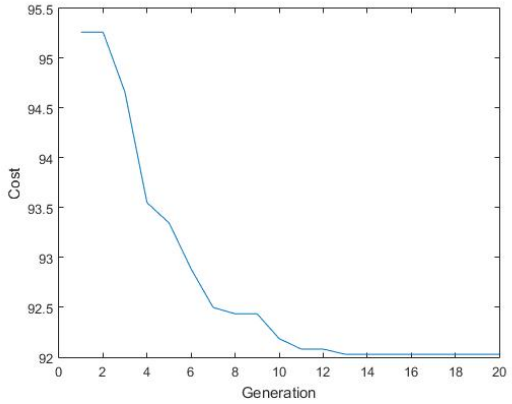


图 4 7 采样点下代数和最小成本关系曲

从结果上来看，遗传代数到 12--15 代，最小成本已经趋于稳定。所以，在实际应用中，只需要选择运行 15 代左右即可得到最优解。

5. 改进方案

5.1 存活概率

在小组实验初期，我们把存活率设置得极低，造成了选择压过大，种群“早熟”。甚至最后存活的个体只有一个。也就是说，过高的选择压不利于区分出优秀的个体和劣质的个体。我们采取的方法是不断尝试调整选择压的大小，寻找到一个主观上合理的结果。但是，能否从理论上寻找到最合理的选择压？通过与老师我交流，我们得知，寻找到这个最合理的结果，需要从统计学层面上分析实际数据。限于我们的知识水平和精力，没能进一步研究。

5.2 拟合方法

最终结果采用三次多项式插值方法得到最优解。这是根据实验结果得到的。我们不清楚为什么要使用该拟合方法，以及是否有更优的拟合方法。

6. 结论

根据多次实验结果，我们保留以下选点方案：

方案 1 取五个采样点（4，16，26，35，48） 成本最优 总成本 84.7483

方案 2 取六个采样点（4，13，30，22，38，49）总成本 87.2135

方案 3 取七个采样点（4，13，21，27，34，43，43，50）总成本 92.0295

从工程应用层面上来看，方案 3 精度最高，方案 1 成本最低。虽然实验结果是方案 1 最优，但是我们认为方案 2 和方案 3 都非常有实际价值。

参考文献：

【1】上海交通大学电子工程系.统计推断在模数数模转换系统中的应用课程.ftp://202.120.39.248

【2】周明，孙树栋 遗传算法原理及其应用 国防工业出版社 1996

附录：

main 函数

```
function best_anw = main()

data=csvread('.\20150915dataform.csv');

pop = 100;
n = 6;

group = Initpop(pop,n);
for k= 1:20
    cost_output = cost(group);
    [group,pop] = select(cost_output, pop, group);
    group = product(pop,n, group);
    group = variation(group);
end

sixth_cost = cost(group);
min_cost = min(sixth_cost);
for j = 1:pop
    if sixth_cost(1,j)==min_cost
        best_anw_index = j;
        break;
    end
end
best_anw = group(best_anw_index, :);
end
```

cost 函数（成本）

```
function output=cost(group)
         [row,col]=size(group);
         output=zeros(1,row);
         for i=1:row
             output(i) = test_ur_answer(group(i,1:col));
         end
end
```

select 函数（自然选择）

```
function [out,pop]=select(cost_output, pop, group)
min_cost = min(cost_output);
cost_output =cost_output - min_cost;
cost_output = cost_output .* cost_output;
lim = max(cost_output);
death = 0;
for i = 1:pop
    if pop < 5
        break;
    end
    isLive = lim*rand();
    if isLive < cost_output(1,i)
        group(i-death,:) = [];
        death = death+1;
        pop = pop-1;
     end
end
out=group;
end
```

product 函数（繁殖）

```
unction out=product(pop, n,group)
for j = 1:pop
    if pop==1
        break;
    end
    mother = randi(pop);
    father = randi(pop);
    while mother==father
        father = randi(pop);
    end
    mother_gene = group(mother,:);
    father_gene = group(father,:);
    for i = 1:n
        isCross = rand();
        if isCross>0.5
            temp = mother_gene(1,i);
            mother_gene(1,i) = father_gene(1,i);
            father_gene(1,i) = temp;
        end
    end
    if test_ur_answer(group(mother,:))>test_ur_answer(mother_gene)
        group(mother,:)= mother_gene;
    end
    if test_ur_answer(group(father,:))>test_ur_answer(father_gene)
```

```
        group(father,:)= father_gene;
    end
end
out = group;
end
```

variation 函数（变异）

```
function out=variation(group)
    [row,col] = size(group);
    for i= 1:row
        for j = 1:col
            isVar = rand();
            if isVar < 0.3
              temp = group(i,j);
              cost_temp = test_ur_answer(group(i,:));
              group(i,j) = randperm(51,1);
              cost_var = test_ur_answer(group(i,:));
              if cost_var > cost_temp
                  group(i,j) = temp;
              end
            end
        end
    end
    out = group;
end
```

Initpop 函数（初始化种群）

```
function group=Initpop(pop,n)
     group=zeros(pop,n);
     for i=1:pop
         group(i,:)=sort(randperm(51,n));
         group(i,1)=1;
         group(i,n)=51;
     end
end
```