统计推断在数模转换系统中的应用

组号：33 号 姓名：武思瑶 学号：5130309538，姓名：陈静仪 学号：5130309539

摘要：本次课题的主要内容是对某产品进行监测，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，对批量生产设计一种成本合理的传感特性校准（定标工序）方案，本文中采用三次样条插值选出六个点，利用遗传算法进行拟合。

关键词：统计推断，三次样条插值，遗传算法

Statistical Inference for the data of Digital-analog conversion system

SBSTRACT: The main content of this topic is to monitor a certain product, input and outputcharacteristics of the sensor components of the monitoring module in a strong nonlinear of batch production design of sensing characteristics calibration of areasonable cost (calibration procedure) scheme, in this passage,we use spline to select six points to analog the profile and we use genetic algorithm to fitness.

Keywords:statisitical,spline,genetic algorithm

1 引言

在某型投入批量生产的电子产品内部有一个模块，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，本课题的主要内容是为该模块的批量生产设计一种成本合理的传感特性校准(定标)方案。[1]

在工程实践中往往会有很多测量工具，这些工具主要由传感器和电子信号接收器组成，为了使批量生产的工具在应用时能够达到一定精度，需要对其用一个科学合理的方法来定标，由于数据较多，人工定标过于费时费力，在非精密仪器的情况下需要能够找到一个省时省力的办法来对这些仪器进行定标，让其达到应有的精度。对于每一个单一的工具，需要根据这个工具产生的某些数据点的反馈进行合理的模拟得出输入-输出关系式，本文中对曲线插值、曲线拟合进行讨论，选取一个较为合理的方法计算输入-输出关系式，在这个过程中启发式搜索起到很大简化问题难度的作用，本文中所采取的的启发式搜索是利用遗传算法对数据进行筛选。

2 评价标准的构建

为评估和比较不同的校准方案，特制定以下成本计算规则。

2.1 单点定标误差成本

$$s_{i,j}=\begin{cases}0 & \text{if } \left|\hat{y}_{i,j}-y_{i,j}\right|\le 0.5\\ 0.5 & \text{if } 0.5<\left|\hat{y}_{i,j}-y_{i,j}\right|\le 1\\ 1.5 & \text{if } 1<\left|\hat{y}_{i,j}-y_{i,j}\right|\le 2\\ 6 & \text{if } 2<\left|\hat{y}_{i,j}-y_{i,j}\right|\le 3\\ 12 & \text{if } 3<\left|\hat{y}_{i,j}-y_{i,j}\right|\le 5\\ 25 & \text{if } \left|\hat{y}_{i,j}-y_{i,j}\right|>5\end{cases} \quad (2\text{-}1)$$

单点定标误差的成本按式（2-1）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。

2.2 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。

2.3 某一样本个体的定标成本

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

对样本 i 总的定标成本按式（2-2）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。

2.4 校准方案总体成本

按式（2-3）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。

$$C=\frac{1}{M}\sum_{i=1}^{M}S_i \quad (2\text{-}3)$$

总体成本较低的校准方案，认定为较优方案。

3 统计算法的理论基础

在本次优化方案的计算中，我们拟采用模拟退火算法和遗传算法进行选择优化，经过学习分析法则，我们决定采用遗传算法来实现。

3.1 遗传算法[2]

遗传算法（ Genetic algorithm，GA）是借鉴生物界自然选择和群体进化机制形成的一种全局寻优算法。

与传统的优化算法相比，遗传算法具有如下优点：

（1）不是从单个点，而是从多个点构成的群体开始搜索；

（2）在搜索最优解过程中，只需要由目标函数值转换得来的适应值信息，而不需要导数等其它辅助信息；

（3）搜索过程不易陷入局部最优点。目前，该算法已渗透到许多领域，并成为解决各领域复杂问题的有力工具。

在遗传算法中，将问题空间中的决策变量通过一定编码方法表示成遗传空间的一个个体，它是一个基因型串结构数据；同时，将目标函数值转换成适应值，它用来评价个体的优劣，并作为遗传操作的依据。遗传操作包括三个算子：选择、交叉和变异。选择用来实施适者生存的原则，即把当前群体中的个体按与适应值成比例的概率复制到新的群体中，构成交配池（当前代与下一代之间的中间群体）。选择算子的作用效果是提高了群体的平均适应值。由于选择算子没有产生新个体，所以群体中最好个体的适应值不会因选择操作而有所改进。交叉算子可以产生新的个体，它首先使从交配池中的个体随机配对，然后将两两配对的个体按某种方式相互交换部分基因。变异是对个体的某一个或某一些基因值按某一较小概率进行改变。从产生新个体的能力方面来说，交叉算子是产生新个体的主要方法，它决定了遗传算法的全局搜索能力；而变异算子只是产生新个体的辅助方法，但也必不可少，因为它决定了遗传算法的局部搜索能力。交叉和变异相配合，共同完成对搜索空间的全局和局部搜索。

遗传算法的基本步骤如下：

（1）在一定编码方案下，随机产生一个初始种群；

（2）用相应的解码方法，将编码后的个体转换成问题空间的决策变量，并求得个体的适应值；

（3）按照个体适应值的大小，从种群中选出适应值较大的一些个体构成交配池；

（4）由交叉和变异这两个遗传算子对交配池中的个体进行操作，并形成新一代的种群；

（5）反复执行步骤 ，直至满足收敛判据为止。

4 拟合方法基本原理

对于某些需要拟合的点，选择一种方法，得到拟合函数$f(x)$，根据最小二乘法，使得拟合模型与实际观测值在各点的残差加权平方和达到最少，此时所求曲线为在加权最小二乘意义下数据的拟合曲线。

曲线拟合方法有多项式拟合法、三次样条插值法。

4.1 三次多项式拟合法[3]

用三次多项式近似拟合曲线

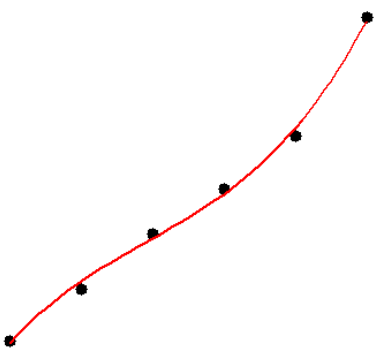

图 3-1 三次多项式近似拟合曲线

使用三次多项式拟合，$f(x)=ax^3+bx^2+cx+d$

实验中，我们利用 matlab 中已有函数（式 4-1）进行拟合：[4]

$$a=polyfit(x0,y0,m) \tag{4-1}$$

其中 x0，y0 为拟合点坐标，m 为拟合多项式次数

多项式在 x 处的 y 值可用 matlab 中已有的函数（式 4-2）进行计算：

$$y=polyval(a,x) \tag{4-2}$$

由于 m=3 时拟合效果较好，平均误差较小，故本课题中我们采用三次拟合。

4.2 三次样条插值法[3]

插值法的基本思想是构造一个简单函数 y=P(x)作为$f(x)$的近似表达式，以P(x)的值作为函数$f(x)$的近似值，而且要求P(x)在给定点x_i与取值相同，即$P(x_i)=f(x_i)$,通常称P(x)为$f(x)$的插值函数，x_i为插值节点。

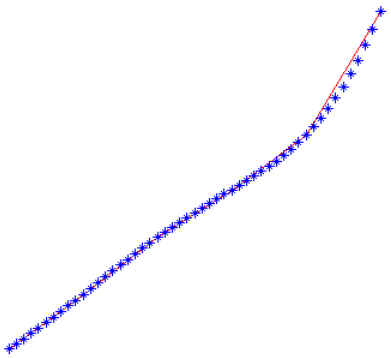

图 3-2 三次样条插值拟合曲线

我们应用 matlab 中已有函数（式 4-3）进行三次插值拟合：

$$y=spline(x0,y0,x) \tag{4-3}$$

其中 x0,y0 为结点坐标，x 为插值点

经过分析比较以及前人的做法，我们得出，三次样条插值算法比三次多项式拟合误差更小，因此在本次课题中，我们选择三次样条插值算法进行拟合。

5 统计推断算法的实现

5.1 问题的引入

通过检测 K 组实验数据，发现每组数据中的 Y-X 大致呈如下图所示关系：

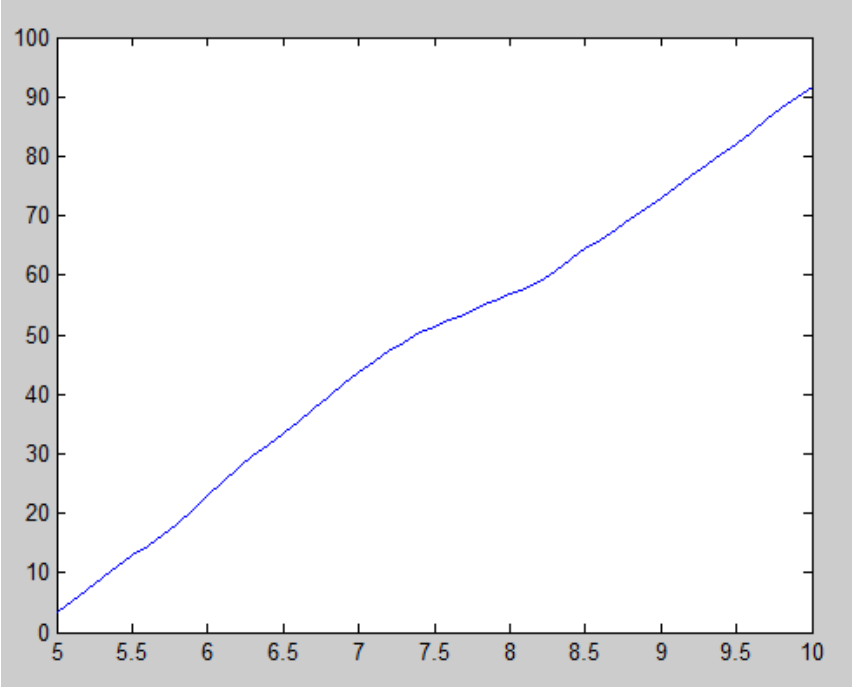

图 5-1 Y-X 关系曲线

$$y_{ij} = f(d_{ij};\ \{x_{ij}\}) + \varepsilon_{ij} \qquad (4\text{-}1)$$

$$i=1,2,\ \ldots,\ K \quad j=1,2,\ \ldots,\ N$$

本问题转化为从 M 个点中选取一定数量的点进行测量，进而推断全部 M 个点的统计推断问题，在本课题中，我们从 M=51 个点中选取一定点，对其进行三次样条插值算法和三次多项式拟合法进行拟合，根据拟合曲线 x 对应的 y 值计算残差，最后将全部样本的残差的平方相加，然后重新选择点进行以上工作，直到选取的点得到的残差的平方和取最优解，即：

$$g = \sum_{所有样本} \sum_{i=0}^{51} \varepsilon_{ij}^2 \qquad (4\text{-}2)$$

所得的 g 为达到的最小值的点的选择方案。

5.2 遗传算法基于 Matlab 的实现

遗传算法的实现步骤：

（1）确定初始条件,种群大小，繁殖次数，交叉互换概率，突变概率，并以六个点的标号作为每一个种群的个体，随机生成一百个个体作为初始种群。

（2）用 4.1 中所提到的代价函数 4-2 的相反数作为适应度函数，并使用三次样条差值法拟合曲线，从而计算每个个体的适应度，称之为 Q。

（3）根据轮盘赌法，使 Q 值小的拥有更多的繁殖机会，通过交叉互换以及突变产生一个新种群。

（4）记录每代中 Q 最小的个体，并继续进行下一代的繁殖，如此继续，直到达到设定的最大代数，此时记录中的最小个体即为我们得到的六个点选取的较优解。

6 实验数据的处理与分析

6.1 通过实验数据确定初始条件

在Matlab遗传算法的实现中，需要确定一些初始化的参数，如编码串长度、种群大小、交叉和变异概率，为保证算法的运行效率和群体的多样性，一般取种群数目为20～100，交叉是产生新个体的基本方法，概率一般取0.4～0.99，变异概率也是新个体产生重要参数，一般取0.0001～0.1，我们选取交叉概率pc=0.85，突变概率pm=0.01，进化代数n=100。

在种群大小的选取上，我们比较了种群数量在50～100大小的种群，计算其截止代数和成本加以比较，得到如下表格：

表6-1 不同种群截止频率与成本比较

Pop	50	60	70	80	90	100
N	65	49	27	22	22	15
Cost	104.1546	105.6638	99.7025	104.5577	98.5516	97.9558

可以看出，随种群数量的增加，截止频率所需的代数越来越少，即种群数量越大，达到优解所需的代数越少，而在种群数量减少时，可能出现不稳定的情况，即有一定的几率发生很大误差，可能需要很多代才能达到满足给定的条件，与应有的规律相悖的结论。

总结以上讨论和所得到的数据，我们决定选取种群个数为 100，进化代数为 100 进行运算。

6.2 特征点的选取

本课题要求选取最少数目的点拟合出代价最小的曲线，若所选取的点数太大会引起成本不必要的增加，若选取点数太小无法准确拟合符合课题要求的曲线我们开始将实验数据分成几个区间，在各个区间内随机选点，通过迭代计算所选点的代价，删除其中代价较大的点，经过 100 次迭代最终所选取的点数稳定在 6 个或 7 个。

表 6-2 不同进化代数与特征点

g	ans									
1	1	10	21	25	29	30	32	40	50	51
5	1	10	21	29	30	32	40	50	51	
11	1	10	21	29	30	32	40	51		
21	1	10	21	30	40	50	51			
28	1	10	21	30	40	51				
50	1	10	21	30	40	51				
100	1	10	21	30	40	51				

Matlab 测得，在所选取的种群条件下，特征点为[1,10,21,30,40,51]。

7 结论

经过以上分析，我们得出遗传算法在本次课题中可以有效的得到最优解，可以在很大程度上缩短计算过程与时间，而在三次插值算法和三次多项式算法的比较中，我们得出三

次插值算法优于三次多项式拟合，最终采用三次插值算法拟合，得到 6 个特征点，分别为[1,10,21,30,40,51]，得到的最优成本为 97.96。

8 致谢

感谢袁焱老师和李老师的指导，对我们的算法和小论文提出了宝贵的意见。

9 参考文献

[1] 上海交大电子工程系. 统计推断讲座 1,2,3 ftp://202.120.39.248.

[2]刘国华，包宏，李文超 用 MATLAB 实现遗传算法程序 北京：北京科技大学

[3]百度百科 词条“三次多项式拟合”，“三次插值拟合”，“遗传算法”

[4]matlab 数据拟合使用教程 ppt

附录：

附录 1：遗传算法主函数（matlab）

```
data=csvread('20141010dataform.csv');
pop=100;
pc=0.85;
pm=0.001;
n=100;
y=zeros(469,51);
y(1:469,:)=data(2:2:938,:);
gene=geneinit(pop);
for g=1:n
    display(g);
    cost=adapt(gene,y,pop);
    gene=select(gene,cost,pop);
    gene=generate(gene,pop,pc);
    gene=mutate(gene,pop,pm);
    display(find(gene(1,:)==1));
end
xx=find(gene(1,:)==1);
assess(xx,y);
```

附录 2：遗传算法主要函数

（1）随机产生初始种群

```
function out = geneinit(pop)

out=round(rand(pop,51)-0.2);
out(:,1)=1;
out(:,51)=1;

end
```

（2）二分法查找

```
function [out] = search(in,s,l,r)

mid=floor((l+r)/2);
if in<=s(mid)
```

```
    if in>s(mid-1)
        out=mid-1;
    else
        out=search(in,s,l,mid);
    end
else
    if in<=s(mid+1)
        out=mid;
    else
        out=search(in,s,mid,r);
    end
end

end
```

（3）交叉保留

```
function out = generate(gene,pop,pc)

for i=2:floor(pop/2+1)
    out=gene;
    mid=floor(rand()*50)+1;
    t=rand();
    if t<=pc
        out(i,1:mid)=gene(pop-i+2,1:mid);
        out(pop-i+2,1:mid)=gene(i,1:mid);
        out(i,mid+1:51)=gene(pop-i+2,mid+1:51);
        out(pop-i+2,mid+1:51)=gene(i,mid+1:51);
    end
end

end
```

（4）自然选择

```
function out = select(gene,cost,pop)

out=zeros(pop,51);
cost0=max(cost)-cost;
```

```
s0=sum(cost0);
s=zeros(pop+1);
s(1)=0;
s(pop+1)=1;
s(2:pop)=sum(cost0(1:pop-1))/s0;
for i=2:pop
    t=rand();
    j=search(t,s,1,pop+1);
    out(i,:)=gene(j,:);
end
sort0=[[1:pop]',cost];
sort0=sortrows(sort0,2);
out(1,:)=gene(sort0(1,1),:);

end
```

（5）变异

```
function out = mutate(gene,pop,pm)

out=gene;
for i=2:pop;
    for j=2:50;
        t=rand();
        if t<=pm
            out(i,j)=~out(i,j);
        end
    end
end

end
```

附录 3：成本计算

（1）计算每个个体平均成本

```
out=zeros(pop,1);
x=5:0.1:10;
for i=1:pop
    c=sum(gene(i,:)==1);
```

```
    pos=find(gene(i,:)==1);
    xx=5+(pos-1)*0.1;
    yy=y(:,pos);
    f=spline(xx,yy);
    dy=ppval(f,x)-y;
    out(i)=12*c+errorcost(dy)/469;
end
min(out)
mean(out)
end
```

（2）单个个体成本误差计算

```
function [out] = errorcost(dy)

t=abs(dy);

t0=sum(sum(t<=0.5));
t1=sum(sum(t<=1))-t0;
t2=sum(sum(t<=2))-t0-t1;
t3=sum(sum(t<=3))-t0-t1-t2;
t4=sum(sum(t<=5))-t0-t1-t2-t3;
t5=sum(sum(t>5));

out=0.5*t1+1.5*t2+6*t3+12*t4+25*t5;

end
```

（3）计算成本

```
function [out] =assess(in,y)

out=length(in)*12;
x=5:0.1:10;
xx=5+(in-1)*0.1;
yy=y(:,in);
f=spline(xx,yy);
dy=ppval(f,x)-y;
out=out+errorcost(dy)/469;
```

```
s=sum(dy.^2);
fid=fopen('answer.txt','a');
fprintf(fid,'position: [ ');
fprintf(fid,'%2d ',in);
fprintf(fid,']    mean_cost: %7f\n\n',out,s);
fclose(fid);

end
```