

# 统计推断在数模转换系统中的应用

50 组 张晨晖 5140309226

摘要：本课题针对某种批量生产的电子产品内部模块的非线性输入输出特性之间的函数关系并进行定标，而不同样本间存在着差异使得每个产品所符合的函数拟合结果相近但不相同，为了最终使用单一定标方式而使得其在精确度符合相应要求情况下减少成本，。我们使用 MATLAB 等计算工具进行程序设计，运用启发式算法之一：遗传算法选择合适的定标点，再使用各类插值及拟合方法来确定产品的特性函数并定标。

关键词：定标，拟合，插值运算，启发式算法，遗传算法

## 1.引言

该课题来源于某种批量生产的电子产品，其内部有一个传感器模块，功能是监测某项与外部环境有关的如温度、压力、光强等的物理量。该监测模块中传感器部件的输入输出呈现出非线性关系。其工作原理见图 1

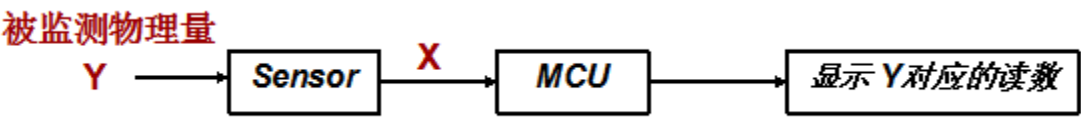


图 1

对于传感器类产品使用前须事先进行定标，然而，在工业生产中，即使是同一批次的产品间也难以保持定标结果一致（见图 2），样本特性曲线呈非线性，不同个体间存在差异因而需要对产品采取采样定标，在满足测量的精确性同时尽可能的减少定标成本

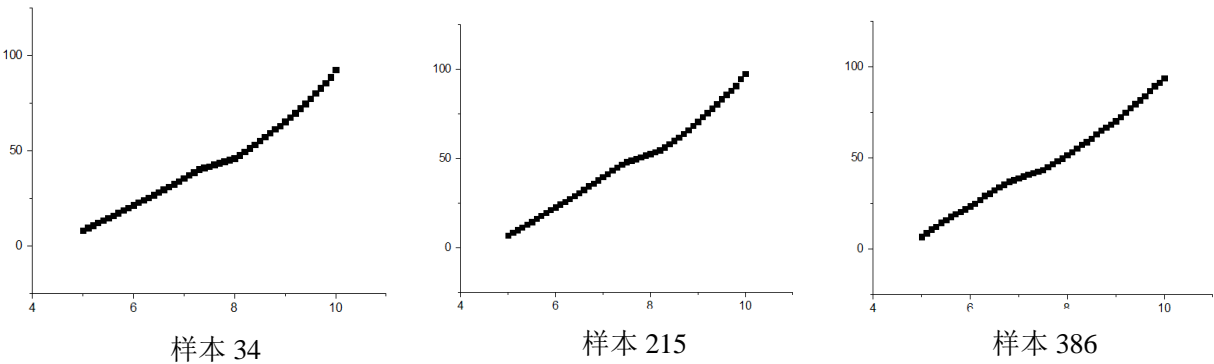


图 2

## 2. 成本计算函数

2.1 单点定标误差成本（下标  $i$  代表样品序号，下标  $j$  代表观测点序号）

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

2.2 对某一样品  $i$  的定标成本（ $N_i$  是该样品定标时所测定的点数，式中第一项为误差成本，第二项为测定成本）

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

2.3 定标方案总成本

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

## 3. 函数插值拟合方式

3.1 多项式拟合

多项式拟合是一种最为基本的拟合方式。对给定数据  $(x_i, y_i)$ ，在给定的函数类  $\Phi$  中，求  $p(x) \in \Phi$ ，使误差  $r_i = p(x_i) - y_i$  为最小，这样求得的函数  $p(x)$  就是拟合出来多项式函数。多项式拟合是一种最普遍，简单易行的拟合方法。一般而言，拟合次数越高，结果越准确，但是其所

用时间呈指数增长。因此，在本实验中选择三次多项式拟合最为合适。我们利用 MATLAB 中已有的函数  $a=\text{polyfit}(x0,y0,m)$ ,其中  $m$  为最高次数。

### 3.2 三次样条插值

三次样条插值法一种非线性插值法，它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。优点为插值所得曲线为光滑曲线；且曲线的导数曲线依然光滑。而且真实曲线的形态，对插值所得曲线偏差影响较小。

使用三次样条插值对确定一个可能解，即 7 个特征点组合  $S=\{s_1, s_2, \dots, s_7\}$  进行拟合。对于非两端点，以四个连续点确定一条三次曲线，但仅在中间两点之间用该三次曲线表示，以此类推，所有非两端点之间均有三次曲线。两端点由端点处三个点用二次曲线拟合。

实际计算时还需要引入边界条件才能完成计算。边界通常有自然边界（边界点的导数为 0），夹持边界（边界点导数给定），非扭结边界（使两端点的三阶导与这两端点的邻近点的三阶导相等）。在 MATLAB 中使用 `spline` 函数进行 3 次样条差值拟和处理。

## 4.启发式算法

### 4.1 遗传算法

遗传算法（Genetic Algorithm）是一类借鉴生物界的进化规律（适者生存，优胜劣汰遗传机制）演化而来的随机化搜索方法。它是由美国的 J.Holland 教授 1975 年首先提出，其主要特

点是直接对结构对象进行操作，不存在求导和函数连续性的限定；具有内在的隐并行性和更好的全局寻优能力；采用概率化的寻优方法，能自动获取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。遗传算法的这些性质，已被人们广泛地应用于组合优化、机器学习、信号处理、自适应控制和人工生命等领域。它是现代有关智能计算中的关键技术。

对于一个求函数最大值的优化问题(求函数最小值也类同)，一般可以描述为下列数学规划模型：式中  $x$  为决策变量，式 2-1 为目标函数式，式 2-2、2-3 为约束条件， $U$  是基本空间， $R$  是  $U$  的子集。满足约束条件的解  $X$  称为可行解，集合  $R$  表示所有满足约束条件的解所组成的集合，称为可行解集合。

遗传算法也是计算机科学人工智能领域中用于解决最优化的一种搜索启发式算法，是进化算法的一种。这种

启发式通常用来生成有用的解决方案来优化和搜索问题。进化算法最初是借鉴了进化生物学中的一些现象而发展起来的，这些现象包括遗传、突变、自然选择以及杂交等。遗传算法在适应度函数选择不当的情况下有可能收敛于局部最优[1]，而不能达到全局最优。

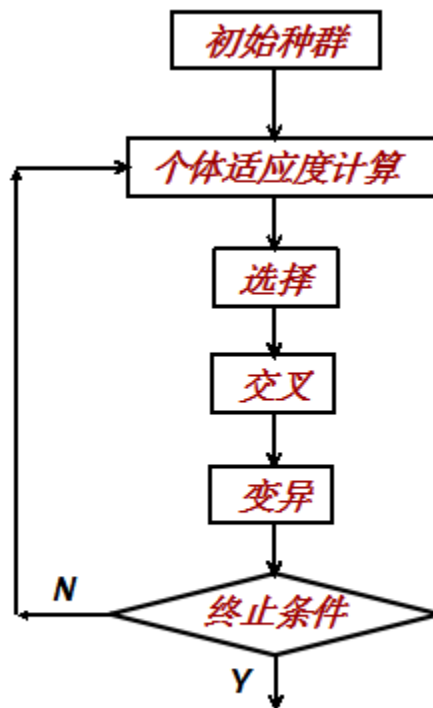


图 3

遗传算法的基本运算过程如下：

a)初始化：设置进化代数计数器  $t=0$ ，设置最大进化代数  $T$ ，随机生成  $M$  个个体作为初始群体  $P(0)$ 。

b)个体评价：计算群体  $P(t)$  中各个个体的适应度。

c)选择运算：将选择算子作用于群体。选择的目的是把优化的个体直接遗传到下一代或通过配对交叉产生新的个体再遗传到下一代。选择操作是建立在群体中个体的适应度评估基础上的。

d)交叉运算：将交叉算子作用于群体。遗传算法中起核心作用的就是交叉算子。

e)变异运算：将变异算子作用于群体。即是对群体中的个体串的某些基因座上的基因值作变动。

群体  $P(t)$  经过选择、交叉、变异运算之后得到下一代群体  $P(t+1)$ 。

f)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，终止计算

## 最终实验方案

### 5.1 取点个数

为保证精确性，取点个数应当尽可能的多；与此同时，为保证成本较低，在一定精确度范围内令取点数最少。综合考虑以上两点，我组选择每组样本选取 7 个数据点，可以最大限度的在一定误差范围内减少成本。

### 5.2 拟合方式

从实验目的来看，三次样条插值法对于图线的拟合相较于多项式拟合更为精确的同时时间性能好，因此，我组决定使用该法进行样本观测值的曲线拟合。

### 5.3 适应度函数

在传统遗传算法中，根据适应度函数值大小决定其生存概率，而本实验中所求的是成本函数的最小值，因此，无法直接使用成本函数作为适应度函数，而应当对成本函数进行函数处理，使得处理后成本最小的个体有最大的概率生存，因此应当使用递减处理函数：为保证适应度高的个体有较大的可能性存活，因此，使用在  $[0, \infty]$  内递减凹函数  $f(x) = -2^x$

## 5.4 遗传算法

### 5.4.1 编码

使用 51 位二进制数表示 51 个样本点，用 0 表示未取该点，用 1 表示取该点来进行初始种群的编码，这样不会像使用单纯二进制转换那样因交叉变异出现超出 51 的点。

### 5.4.2 参数设定

在本实验中，经过多次程序运行，可得最优解参数：遗传代数 75，初始种群个体数 100，变异概率 0.08

## 6.实验结果

取点：3 8 15 21 31 42 44 49          成本 111.58

## 7.鸣谢

感谢袁焱老师在该实验中给予我的指导与帮助，我得以顺利完成本次实验

## 8.参考资料

(1) 上海交通大学电子工程系.统计推断在数模转换系统中的应用课程讲义

<ftp://202.120.39.248>

(2) 百度百科——遗传算法：

[http://baike.baidu.com/link?url=PdO6fXCAnyBdQY8hgZNf0-1y1VUe5K2LvyYpTr1vWrAytatM6rqEVS4mU6tNfxb4LCN2I43qUCtF6K\\_PJ4F\\_Bq](http://baike.baidu.com/link?url=PdO6fXCAnyBdQY8hgZNf0-1y1VUe5K2LvyYpTr1vWrAytatM6rqEVS4mU6tNfxb4LCN2I43qUCtF6K_PJ4F_Bq)

附：遗传算法代码

intersect.m

function intersect

```
    global p n;
    for t=1:(n/2)
        k=randi([10,40],1,1);
        temp(1:k)=p(t,1:k);
        p(t,1:k)=p(t+(n/2),1:k);
        p(t+(n/2),1:k)=temp(1:k);
    end
```

end

fitness.m

function anss=fitness(method)

```
    global y mini minp;
    x=5.0:0.1:10.0;
    b=0; k=1;
    x1=zeros(1,51);
    y1=zeros(400,51);
    while k<=51
        if method(k)==1
            b=b+1;
            x1(b)=x(k);
            y1(1:400,b)=y(1:400,k);
        end
        k=k+1;
    end
    sum=0;
    for s=1:400
        h=12*b;
        yb=interp1(x1(1:b),y1(s,1:b),x,'spline');
        for t=1:51
            h=h+calc(y(s,t),yb(t));
        end
        sum=sum+h;
    end
    sum=sum/400;
    anss=1/sum;
    if sum<mini
```

```

        mini=sum;
        minp=method;
    end
end

```

```

function h1=calc(ya,yb)
    if abs(ya-yb)<=0.4
        h1=0;
    elseif abs(ya-yb)<=0.6
        h1=0.1;
    elseif abs(ya-yb)<=0.8
        h1=0.7;
    elseif abs(ya-yb)<=1
        h1=0.9;
    elseif abs(ya-yb)<=2
        h1=1.5;
    elseif abs(ya-yb)<=3
        h1=6;
    elseif abs(ya-yb)<=5
        h1=12;
    else
        h1=25;
    end
end

```

```

start.m
function start
    global p n;
    t=0; n=100;
    P=8/51;
    dict=[0,1;1-P,P];
    p=zeros(n,51);
    while t<n
        temp=randsrc(1,51,dict);
        if fitness(temp)>0.0004
            t=t+1;
            p(t,1:51)=temp(1:51);
        end
    end
end

```

```
end
```

```
clear;  
load('var.mat');  
global p y mini minp;  
T=75;  
mini=1e100;  
start;  
for t=1:T  
    selection;  
    intersect;  
    variation;  
end  
disp(mini);  
disp(minp);
```

variation.m

```
function variation  
    global p n;  
    P=0.1;  
    dist=[1,0;P,1-P];  
    for s=1:n  
        for t=1:51  
            flag=randsrc(1,1,dist);  
            if flag==1  
                p(s,t)=1-p(s,t);  
            end  
        end  
    end  
end
```

selection.m

```
function selection  
    global p n;  
    sum=0;  
    fitn=zeros(1,n);  
    for s=1:n
```



```

        fitn(s)=fitness(p(s,1:51));
        sum=sum+fitn(s);
    end
    dist=[1:n;fitn(1:n)/sum];
    temp=zeros(n,51);
    for s=1:n
        col=randsrc(1,1,dist);
        temp(s,1:51)=p(col,1:51);
    end
    p=temp;
end

```

%%%%%%%%%% 答案检验程序 2015-11-04 %%%%%%%%%%%

```

my_answer=[ 3,8,15,21,42,44,49 ];%把你的选点组合填写在此

```

```

my_answer_n=size(my_answer,2);

```

% 标准样本原始数据读入

```

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;


% 定标计算

index_temp=logical(my_answer_gene);

x_optimal=x(:,index_temp);

y0_optimal=y0(:,index_temp);

for j=1:nsample

    % 请把你的定标计算方法写入函数 mycurvefitting

    y1(j,:)=mycurvefitting(x_optimal(j,:),y0_optimal(j,:));

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(nsample,1)*my_answer_n;
```

```
cost=sum(si)/nsample;
```

```
% 显示结果
```

经计算，你的答案对应的总体成本为 **111.58**