

# 统计推断在数模转换系统中的应用

组号 23 姓名 李抒昊 学号 5140309230 姓名 朱乐岩 学号 5140309228

**摘要：**假定有某型投入批量试生产的电子产品，其内部有一个模块，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。本课题要求为该模块的批量生产设计一种成本合理的传感特性校准（定标工序）方案。

**关键词：**定标，插值拟合，遗传算法

## Application of Statistical Inference in AD&DA Inverting System

**ABSTRACT:** Assuming that there're some electronic products taken into trial. We are focusing on one of its internal modules. The module has the function to monitor a physical quantity related to the outside environment (which may be temperature, pressure, light intensity, etc). The character of the input and output of sensor in the monitoring module is obviously nonlinear. The subject of this thesis is to design a kind of calibration procedure with a reasonable cost for the mass production of the module.

**Key words:** Statistic Interference, Sensing characteristics, Fit, Genetic Algorithm

## 1 引言

首先快速把课题的目标解释一遍。

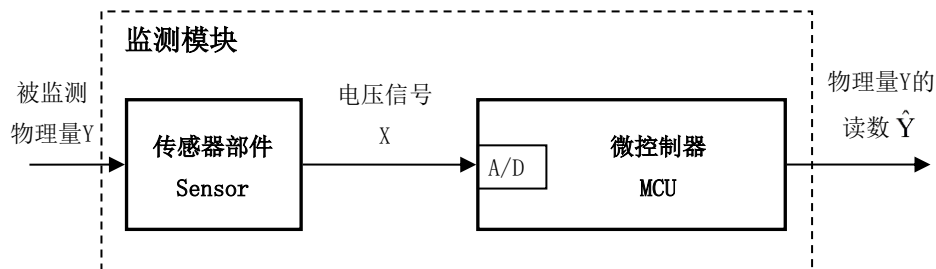


图 1-1 监测模块组成框图

监测模块的组成框图如图 1。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；传感器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）。<sup>[1]</sup>

所谓传感特性校准，就是针对某一特定传感部件个体，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。

定标成本分为测定成本和误差成本两项,测定点的个数多以及最终结果误差大都会导致测定成本变高。而我们的最终目的是得到最小的定标成本。

## 2 解决过程

### 2.1 数学模型的选取

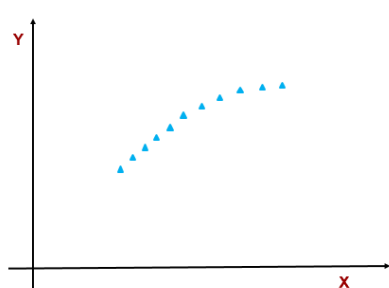


图 2-1 未用插值拟合方法

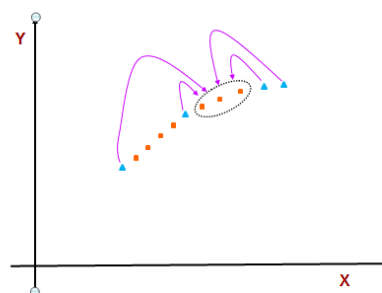


图 2-2 使用插值拟合方法

在设计过程中我们需要用到拟合插值的方法。如图 2-1 所示,这张图里面我们精确测试了 11 个样本点,显然这样的方法得到的结果是零误差的,但是它的测量成本过高。在图 2-2 中运用了拟合插值的方法。测定了四个点,然后作拟和线条,然后求得在待测点的  $y$  值。

我们尝试用拟合和插值两种方式来获取未测量点的估计值。<sup>[2]</sup>

拟合和插值这两种方式都是通过已知点来估计未知点的函数值的工具,其不同点是:用拟合方式拟合出的函数曲线不一定过样本点,存在残差。用插值方式得出的函数曲线必定过给定的样本点。

我们组在探索的过程中即用了拟合方式也用了插值的方法。拟合采用了多项式拟合,其中线性拟合显然不会是最优的,所以我们组直接跳过了线性拟合,而将多项式拟合的次数设定为 2, 3, 4. 而插值法我们则选用了比较通用的三次样条插值法 (Cubic spline)。

### 2.2 取点方式的探索

为了选取合适的样本点来拟合或插值,我们需要确定取点方案。

#### 2.2.1 穷举法

最好想到的就是穷举法,就是将每一种取点方式都一一计算一遍,这样的方法不会漏过任何一种取点方式,得到的结果也一定是最优解。但是穷举法存在一个非常严重的弱点,那就是算法复杂度太高,我们来计算一下,以取 8 个点为例,总共有  $C_{51}^8$  种情况,然后取点个数可以是 2,3,4,...,51,这样一来情况总数就是  $C_{51}^2 + C_{51}^3 + C_{51}^4 + \dots + C_{51}^{51}$  约是  $2^{51}$ ,这样的数量级是计算机所不能承受的,它无法在合理的时间内得到我们想要的结果,所以这种方式不可取。

#### 2.2.1 遗传算法

在尝试穷举法失败后,我们采用遗传算法来改进刚才的思路。遗传算法 (GA, Genetic Algorithm) 是一种模拟达尔文进化理论的自然选择和基因遗传的一种计算模型,是一种通过模拟自然界生物世代繁殖来搜索最优解的方式。我们都知道基因变异是无向的而自然选择是有向的,我们就是利用这一点,使得我们不需要枚举所有的取点方式,就可以让结果朝着最优解的方向去进化。

## 3 遗传算法

### 3.1 遗传算法的步骤

(a) 初始化种群(initialize): 我们随机生成 N 个种群, 种群中的每一个个体都是用一个长为 51 的二进制数组来表示的, 其中 51 位的每一位上 1 表示相应点有取, 0 表示相应点没有取。

(b) 选择(selection): 对种群中的每一个个体, 计算其适应度, 适应度表示的是这个个体的优劣程度, 适应度越高, 个体越优秀。我们适应度函数的选取直接采用成本的倒数。计算出所有个体的适应度后, 将其全部加起来, 得到种群的总适应度, 每个个体适应度占总适应度的比例成为个体的生存概率。按照生存概率进行选择, 同样选择 N 次, 这一百个个体(可能重复)生存了下来。

(c) 交叉(crossover): 对选择后的个体执行交叉运算, 我们采用单点交叉, 即随机交叉点, 然后将两个个体交叉点前的部分互换。

(d) 变异(mutation): 对每个基因点都有一定的概率变异, 我们设定变异概率 P, 每个点有概率 P 取反。

(e) 繁衍: 经过选择、交叉、变异后得到的新种群就是原种群的下一代, 我们设定繁殖代数 T, 即重复以上步骤 T 次, 将在这个过程中得到的最优解输出。

### 3.2 遗传算法的不断改进

我们首先选取遗传代数  $T=50$ ,  $P=0.01$  种群规模是  $N=50$  个, 使用三次样条插值方法进行运行, 最后运行出来的最小适应度是 105.43, 取点方案是 4, 5, 13, 24, 34, 41, 50。我们发现, 事实上以取点成本为 12 来算, 最优解的取点规模无论如何不可能超过 8, 也就是说我们可以改进算法使得取点个数在 8 个点附近。我们改写了初始化函数, 将其从完全随机变成依概率随机, 我们将概率设置为  $7/51$ , 也就是说平均每个个体有 7 个点被选取, 而初始化的实际结果也在 7 个点附近。我们这样做的目的是优化初始种群的质量, 使得初始种群就一定程度上是优秀的, 这样有利于寻找最优解。

我们又执行了一次代码, 发现最优值缩小到了 101.43, 我们又在这个代码的基础上扩大的种群规模, 从 50 扩大到 100, 这样做的目的是增加个体多样性, 利于自然选择。我们还调大了变异概率 P, 从 0.01 调至 0.08, 这样做的目的是为了使得程序不容易陷入局部最优解。

### 3.3 各种拟合插值方式的尝试

基于以上的代码, 我们对代码中的拟合插值方式进行更改, 尝试了二次拟合、三次拟合的情形。

## 4 实验结论

|         |                           |        |
|---------|---------------------------|--------|
| 二次多项式拟合 | 3, 15, 18, 29, 35, 41, 51 | 297.18 |
| 三次多项式拟合 | 3, 11, 16, 21, 31, 45, 50 | 142.48 |
| 三次样条插值  | 3, 10, 19, 26, 34, 45, 50 | 96.08  |
| 立方插值    | 2, 12, 17, 24, 37, 45, 49 | 106.28 |

## 5 参考文献

- [1] “统计推断”课程设计的要求 V2.2 2015-9-22
- [2] 统计推断讲座 2\_问题的提出和基本求解思路

附录： matlab 代码

function crossover

%交叉

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

end

function anss=fitness(method)

%适应度函数计算

```
global y mini minp;
x=5.0:0.1:10.0;
num=0; pos=1;
x1=zeros(1,51);
y1=zeros(400,51);
while pos<=51
    if method(pos)==1
        num=num+1;
        x1(num)=x(pos);
        y1(1:400,num)=y(1:400,pos);
    end
    pos=pos+1;
end
sum=0;
for s=1:400
    h=12*num;
    yb=interp1(x1(1:num),y1(s,1:num),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
```

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
function initialize
%初始化种群
    global p N;
    t=0; N=100;
    P=7/51;
    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
clear;
load('var.mat');
global p y mini minp tt pp;
T=30;
mini=1e100;
initialize;
for tt=1:T
    selection;
    crossover;

```

```

        mutation;
    end
    plot(pp);
    disp(mini);
    disp(minp);
    function mutation
    %变异
        global p N;
        P=0.08;
        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
end
function selection
%选择
    global p N tt pp;
    sum=0;
    fitn=zeros(1,N);
    for s=1:N
        fitn(s)=fitness(p(s,1:51));
        sum=sum+fitn(s);
    end
    pp(tt)=sum/N;
    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
end

```