

统计推断在数模转换系统中的应用

组号 11 杨雲旭 5140309286 李成源 5140309285

摘要: 统计推断在数模转换系统是一种非常重要的方法,通过快速、有效地对样品进行定标,从而获得其输入信号和输出信号的关系。本课题中面对 51 组庞大的数据,一一进行校准成本无疑太大,所以选出一种合适的算法和拟合方法是成功校准的必备条件。

关键词: 模拟退火算法、遗传算法、三次样条插值法、多项式拟合法

Abstract: Statistical inference in analog-to-digital conversion system is a very important method, the samples through quickly and effectively to calibration, to obtain the relationship between the input signal and output signal. This topic in the face of 51 large set of data, each calibration cost is too big, so choose a suitable algorithm and the fitting method is a prerequisite for successful calibration.

Key Words: Simulated annealing algorithm, genetic algorithm, cubic spline interpolation method, polynomial fitting

1. 引言

1.1 课题的提出

在实际生产中,由于每一个器件存在个体差异,所以在产品出厂时需要对每一个特定的器件进行校准和标定。但是若对每个器件都进行完全精准的定标,成本则很大。所以,这个课题就是研究如何在少测量点的情况下还能较精准的拟合传感器部件监测对象 Y 与传感器部件的输出电压信号 X 的函数关系,使误差尽可能地小。

1.2 课题研究的主要步骤

- (1) 初步分析样本数据。
- (2) 选择一种拟合插值方法,也就是选择数学表达形式。
- (3) 选择合适的搜索算法,测算成本并记录,确定7个数据点。
- (4) 分析结果,更换拟合方法。
- (5) 重复(2)(3)(4),将成本最低的方案(拟合+取点)作为最终结果。
- (6) 讨论 6、7 个取样点时结果如何。

2. 定标准确度评价方法

1. 单点定标误差成本:

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

2.单点测定成本: $S_i = \sum_{j=1}^{51} s_{i,j} + q \cdot n_i \quad (q=12)$

3.校准方案总成本: $C = \frac{1}{M} \sum_{i=1}^M S_i$

总成本较低的校准方案，认定为较优方案。

3. 插值或拟合方法

3.1 三次样条插值法

3.1.1 概述

三次样条插值（简称 **Spline** 插值）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。

3.1.2 优缺点分析

优点：所得曲线为光滑曲线，且数据点的估计值等于观察值，比较吻合真实 X-Y 曲线。

缺点：运算量很大，速度慢。

3.1.3 MATLAB 实现

`Y1=interp1(X,Y,voltageX(i,:), 'spline') ;`

其中X和Y分别为数据点的X值和Y值，列表voltageX(i,:) 为所有数据点的电压信号X，Y1为所有数据点Y的估测值列表。MATLAB中将非扭结边界条件作为默认的边界条件。

3.2 分段三次 hermite 插值法

3.2.1 概述

不少实际的插值问题不但要求在节点上的函数值相等，而且还要求对应的导数值也相等，甚至要求高阶导数也相等，满足这种要求的插值多项式就是埃尔米特插值多项式。

3.2.2 优缺点分析

优点：运算速度快，易于理解。拟合结果可能产生极其符合实验点的拟合函数。

缺点：所需要提供信息太多，光滑度不高（只有一阶导数连续）。

影响较大。

3.2.3 MATLAB 实现

SubHermite

YI=interp1(X,Y,XI, 'pchip')

3.3 插值或拟合方法的选定

对于拟合方法的选取，我们是通过实验结果对比得出来的。分别用两种方法来拟合结果，观察他们的总成本和所需时间，从而得出最佳拟合方法。

编号	运行结果	成本	时间
1	7, 15, 19, 24, 30, 38, 48	93.1551	248.15191
2	6, 14, 20, 23, 28, 37, 49	92.8955	247.51919
3	7, 14, 18, 23, 29, 36, 50	93.4516	247.89151

表 3.3.1 三次样条插值法的 7 点模拟退火结果

编号	运行结果	成本	时间
1	7, 14, 17, 23, 29, 37, 49	91.1887	245.96425
2	7, 15, 18, 23, 29, 37, 50	91.0852	245.53644
3	6, 15, 17, 24, 30, 38, 49	91.8155	246.01515

表 3.3.2 分段三次 hermite 插值法的 7 点模拟退火结果

从上述表格可以看出，分段三次 hermite 插值法得到的结果平均成本较低，时间较短，故选用这种方法。

4. 启发式搜索

对于7个观测点的选取，可以通过搜索算法来完成。易知观测点的选取共有 C_{51}^7 ，即约1.15亿中选择，其复杂度为 $O(N!)$ ，所以这是一个NP问题。采用“暴力穷举”法显然不适合此问题，所以我们采用启发式搜索来解决这个问题。

4.1 遗传算法

遗传算法（GeneticAlgorithm）是一类借鉴生物界的进化规律（适者生存，优胜劣汰的遗传机制）演化而来的随机化搜索方法。

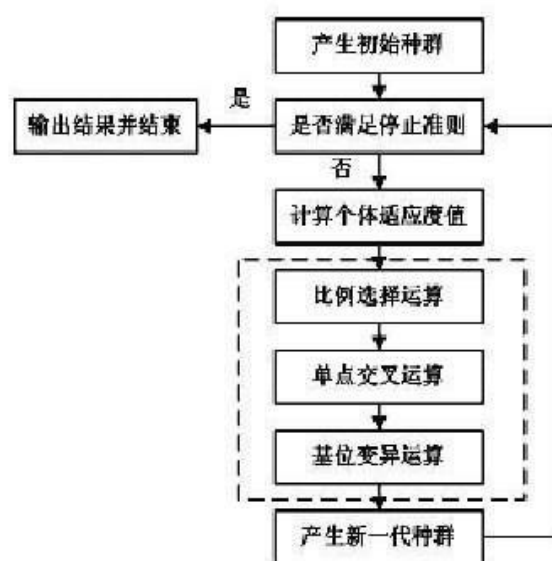


图 3 遗传算法流程图

首先随机选取亲代样本，对其进行评估，筛选掉适应度较低的样本，留下适应度较高的样本，并对适应度较高的样本进行交叉配对，产生第二代，再从整体中随机取得另一部分的样本加入第二代。通过不断产生下一代，适应度高的样本逐渐留下来，适应度低的样本被淘汰，最后得到适应度最高的一组。

下面是遗传算法基本运算过程：

1) 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 $P(0)$ 。

2) 个体评价：计算群体 $P(t)$ 中各个个体的适应度。

3) 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。

4) 交叉运算：将交叉算子作用于群体。所谓交叉是指把两个父代个体的部分结构加以替换重组而生成新个体的操作。遗传算法中起核心作用的就是交叉算子。

5) 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 $P(t+1)$ 。

6) 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。

不过由于遗传算法难以控制最优解的点数，不利于拓展试验的进行，并且其代码难度较高，实现起来较为繁琐，故我们不将其作为本实验选用的算法。

4.2 模拟退火算法

模拟退火算法来源于固体退火原理，是一种基于概率的算法，将固体加温至充分高，再让其徐徐冷却，加温时，固体内部粒子随温升变为无序状，内能增大，而徐徐冷却时粒子渐趋有序，在每个温度都达到平衡态，最后在常温时达到基态，内能减为最小。相比于传统算法，模拟退火算法可以轻易地控制测试点的个数，便于对比实验的进行。另，由于该代码运行时间短，效率高，并且代码实现简单有效，因此我们选用模拟退火算法作为本次课程探究的使用代码。

4.3 模拟退火算法的分析及其实现

模拟退火算法（SA）来源于固体退火原理，将固体加温至充分高，再让其徐徐冷却，加温时，固体内部粒子随温升变为无序状，内能增大，而徐徐冷却时粒子渐趋有序，在每个温度都达到平衡态，最后在常温时达到基态，内能减为最小。根据 Metropolis 准则，粒子在温度 T 时趋于平衡的概率为 $e(-\Delta E/(kT))$ ，其中 E 为温度 T 时的内能， ΔE 为其改变量， k 为 Boltzmann 常数。用固体退火模拟组合优化问题，将内能 E 模拟为目标函数值 f ，温度 T 演化成控制参数 t ，即得到解组合优化问题的模拟退火算法：由初始解 i 和控制参数初值 t 开始，对当前解重复“产生新解→计算目标函数差→接受或舍弃”的迭代，并逐步衰减 t 值，算法终止时的当前解即为所得近似最优解，这是基于蒙特卡罗迭代求解法的一种启发式随机搜索过程。退火过程由冷却进度表(Cooling Schedule)控制，包括控制参数的初值 t 及其衰减因子 Δt 、每个 t 值时的迭代次数 L 和停止条件 S 。

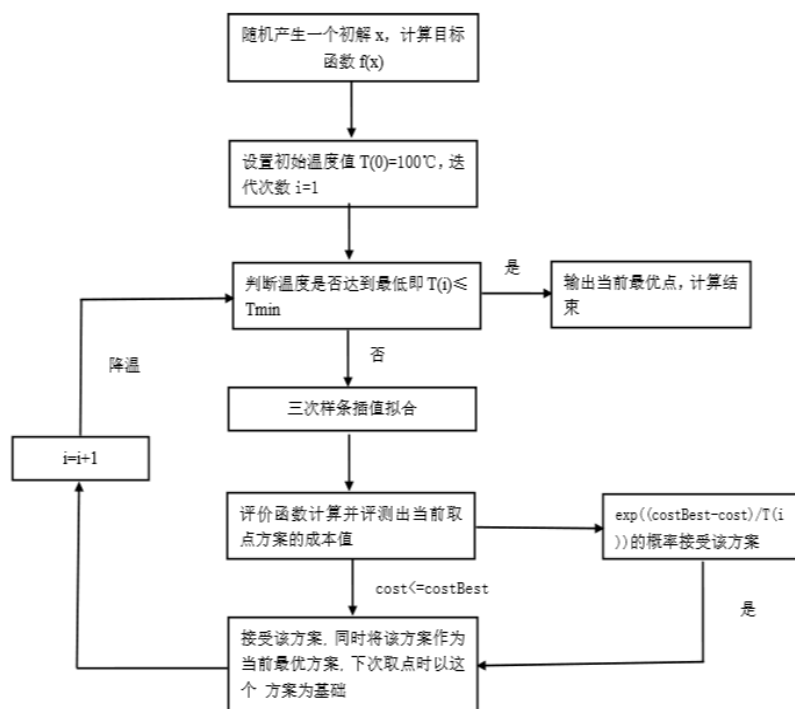


图 4 模拟退火算法流程图

通过 `xlsread` 函数读入数据并分别取出 x 和 y , 通过随机函数随机排序并选取前七个点作为初始点选取, 之后再次排序并取点。设定退火的初始温度 $T(0)=100^{\circ}\text{C}$ 和最低温度 $T_{\min}=0^{\circ}\text{C}$ 以及 i 变化时降温的幅度 $\Delta T=T(i+1)-T(i)=1$ 摄氏度, 构成模拟退火算法作为主体的大循环在循环内前部分。通过样条插值拟合, 得到每点处拟合曲线的值, 同时算出拟合值与实验值的差值, 用评价函数计算当前取点方案的成本值, 并通过比较若此成本值 cost 小于当前最小的成本值 costBest , 则接受该方案并将其作为当前最有方案, 下一次的取点以这个方案作为基础; 若此方案成本值大于最优方案, 那么有 $\exp((\text{costBest}-\text{cost})/T(i))$ 的概率接受该方案。然后往复循环, 不断降温至最低温后输出最优解及其花费的时间。

4. 改变观测点数目的研究

编号	运行结果	成本	时间
1	3, 13, 23, 29, 38, 50	87.3242	243.280158
2	2, 12, 21, 29, 38, 49	87.7875	240.706421
3	4, 13, 25, 33, 44, 50	90.7135	239.835940

表 4.1 分段三次 hermite 插值法的 6 点模拟退火结果

通过以上的数据表格得知, 当选取 6 个点的时候总成本最少, 时间最少, 所以产生 6 个最优特征点是最好的情况。

5. 结论

在 MATLAB 下我们采用模拟退火算法和分段三次 hermite 插值法, 经过很多次的运行, 发现产生 6 个点的情况占了绝大多数, 5 个点和 7 个点的情况也会出现。这也同样说明了 6 个点是成本最少, 最高效的最优方案。

在类似的工业定标中，我们可以同样采用模拟退火算法和分段三次 hermite 插值法得到最优解，减少工作量并同时保证工作的准确性，从而大大地节约了成本。

6. 参考文献：

【1】“统计推断”课程设计的要求 V2.2 2015-9-22

<ftp://202.120.39.248>.

【2】三次样条插值_百度百科

http://baike.baidu.com/link?url=bZefaKxXPmng2K5GyD42brivz4vPJS1qP1GUgnWTRoUR1m_uIRyH24LwJhD84X_jK1X2v16YMEgW6HohLhcKsq

【3】模拟退火算法_百度百科

http://baike.baidu.com/link?url=1IWsqcd6wnU_CjVniAQN9ESOWSxtz1LmFKpvXNv4yC7s2XoLxuwaGrCXA_3Bj8Gc4Ndayo6tVJGKVBZGcoiMnK

【4】遗传算法_百度百科

http://baike.baidu.com/link?url=C2aNeGLv7H4rxjlIA5Ztm-Q5Gp6ncgvf7qmHC1ZAhd1FvhbywgawBCK-zWldVrCyRI_gFE8tJ0pkShpSNGW_Qq

【5】代码方面：统计推断-第 06 组(谢昊男)课程设计报告终稿

附录：

%将下列三个程序放在一个文件夹，运行主程序即可

%fitness

function Result= fitness(~,Result ,Gene)

global originY;

global x;

global tempMaxChosenY;

tmp=12*51+400*51*25;

m=1;

[Xtemp,Ytemp]=find(Gene(m,:)~=0);

count=length(Xtemp);

tempX=Ytemp.*0.1+4.9;

ChosenY=originY(:,Ytemp);

%由于取点之后你和方式优劣各异，下面写出了三次 i 样条插值、分段三次 hermite 插值法

%并有运行结果知后者更佳

%calculateY=spline(tempChosenX,ChosenY,x); %采用三次样条插值方法

calculateY=pchip(tempX,ChosenY,x); %分段三次 hermite 插值法

%成本计算函数

errorCount=abs(calculateY-originY);

le0_4=(errorCount<=0.4);

le0_6=(errorCount<=0.6);

le0_8=(errorCount<=0.8);

le1_0=(errorCount<=1);

le2_0=(errorCount<=2);

le3_0=(errorCount<=3);

le5_0=(errorCount<=5);

g5_0=(errorCount>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);

Result(1,m)=sum(si)/400+12*count; %计算误差成本和测定成本

tempM=tempX;

tempMaxChosenY=Ytemp;

tempMaxChosenY;

end

%main

tic;

clear all;

clc;

minput=dlmread('20150915dataform.csv');

M=800;

N=51;

samplenum=M/2;

```

npoint=N;
global x;
x=zeros(1,npoint);
global originY;
global tempMaxChosenY;
average=0;
originY=zeros(samplenum,npoint);
calculateY=zeros(400,51);
M=1;
Result1=zeros(1,1);
Result2=zeros(1,1);
for i=1:samplenum
    x(1,:)=minput(i*2-1,:);
    originY(i,:)=minput(2*i,:);
end
tempMin=12*51+400*51*25;%×îĐ;³É±¼ÁÛÊ±Öµ£¬²¢Éè¶··³ÖÖµ
Gene1=round(1*rand(M,51));%Ëæ»úÑ;¶··µÚÒ»´úÈ;µã¼¯°İ
T=50000;%³ÖÊ¼Éè¶··µÄİÂ¶È;
generation = 1;
Record1=[];
Record2=[];
Record3=[];
while T>0.1%ÖÖÖ¹İö¼þ£¬İÂ¶È¼öĐ;µ¼×îĐ;Öµ
    flag2=1;
    k=0;
    Record2=[];
    for j=1:90
        flag = 0;
        Gene2 = bianyi(Gene1);%±äÒìµÄµ¼ĐÂ¼â
        Result1 = fitness(1,Result1, Gene1);
        Record2 = [Record2,sum(Result1)];%Çó°İ
        Result2 = fitness(1,Result2, Gene2);%´øÈèÊÊÓ;¶È¯ÊÝÇóÆä³É±¼
        t=Result2-Result1;%³É±¼²îÖµ£¬ÓÃÓÚĀĐ¶İİÓĀÁÓ
        if t<0||rand<exp(-
            t/(T*0.001)) %ĐÂ¼âÓĀÓÚÔ-¼â»òÒÒ»¶¶·µÄĐ;¼ĀÊ¼ÓÊÛĐÂ¼â
                Gene1 = Gene2;
                flag=1;
                if Result2<tempMin
                    tempMin=Result2;
                    Recorder=tempMaxChosenY;
                end
            end
        end
    if flag==1
        k = 0;

```



```

else
    k = k+1;
end
if k>40
    flag2=0;
    break;
end
end
average=sum(Record2)/90;%Çó³ùÖµ
if flag2==0
    break;
end
T%Êä³öµ±Ç°Ê±¼ä
T=T*0.8;%¿ØÖÆÍË»ð£´´µîÂ)¿ìÂÝ£¬Óëµü´ú´îÊÝÓÐ¹Ø
tempMin%Êä³öµ±Ç°×îÓÀ½â
Record1=[Record1,tempMin]; % Record1¼ÇÃ¼Ã¿Ò»¸öÎÂ¶ÈÖÐµÃ×îÐ;¶´±êÖµ
Record3=[Record3,average]; % Record3¼ÇÃ¼Ã¿Ò»¸öÎÂ¶ÈµÃÆ½³ùÖµ
Recorder%Êä³öµ±Ç°Èµã
generation%Êä³öµ±Ç°´úÊÝ
generation=generation+1;
end
bar(Record1) %»-³ö×îÐ;¶´±êÖµ°ÍÆ½³ùÖµµÃÍ¼ÐÎ
axis([0 60 150]);%°á×ø±êÓëµü´ú´îÊÝÓÐ¹Ø;Ç×Ý×ø±êÓë×îÓÀ½âµÃ±ä»¯ÓÐ¹Ø
hold on;
plot(Record3)
axis([0 60 75 255]);
toc;

%bianyi
function m2 = bianyi( m2 )
    for i=1:51
        if (rand<0.03)%小概率变异
            m2(1,i) = 1-m2(1,i);
        end
    end
end
end

```