

统计推断在数模转换系统中的应用

组号 21 姓名 夏煜 学号 5140309132 姓名 吴子镜 学号 5140309130

摘要：本报告主要展示统计推断在模数、数模转换系统中的应用。以上海交通大学电子系实验所测得的共计 400 组输入信号 X 与输出 Y 的关系，运用一定的数理统计方法，经过特征点选取、算法研究、拟合比较等一系列过程，借助 Matlab 建立函数曲线模型，终实现以少量数据反映整体系统特性的效果，从而有效地降低工程设计上的成本，提高效率。

关键词：特征点，曲线拟合，样条插值拟合，遗传算法

The application of statistical reference in the system of AD-DA conversion

ABSTRACT: This report is mainly about the application of statistical inference processing the data in A/D & D/A transformation. By the method of mathematical statistics, with procedures of characteristic point selecting, Algorithm Researching and fitting comparing, relation curve model between the input and the output can be established by Matlab. Finally, we can find a way by which an entire system character will be reflected with less data measuring, thus effectively reduce the engineering design of the cost and improve efficiency.

Key words: feature points, polynomial fitting, Spline difference fitting, genetic algorithm

1 引言

随着科技的发展，传感器得到广泛得应用，要在确保质量的前提下大批量生产传感器，就需要对样品进行单点测定。我们研究的问题寻找校准工序的优化方案。通过预先抽样研究多个样品的曲线特性分析，发现样品的曲线有如下特点：

- (1) 个体产品的特性曲线形态相似但两两相异；
- (2) 特性曲线都是单调递增的；
- (3) 每条特性曲线大致可以分为首（左），中，尾（右）三段，三段都不是完全线性的，有一定的弯曲度（随机）；
- (4) 中段的斜率小于首段和尾段的斜率，且中段的起点位置和长度都带有随机性。

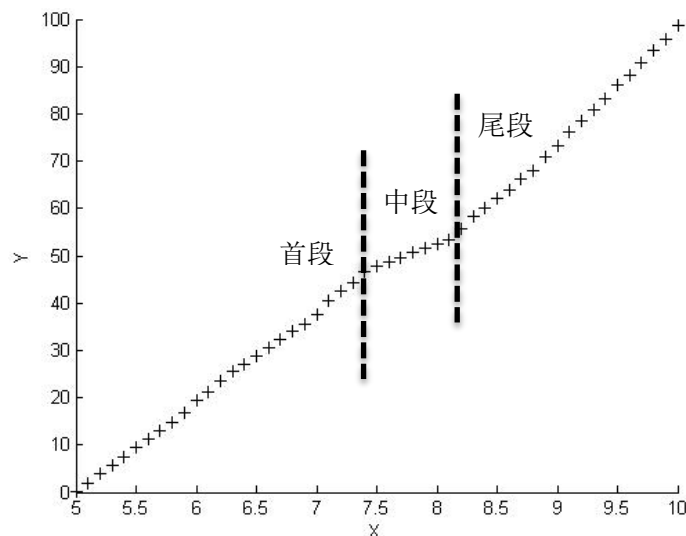


图 1 传感特性图示

所以我们需要选择一种拟合（或插值）方法，并且选择一种取点方案，依据从样本库获取的单点测定数值，按照选择的拟合方法进行拟合并逐一完成定标，根据定标结果，结合原始数据测算出成本记录，最后方案对比将成本最低的方案作为最终结果。

考虑到拟合效果和运行时间，我们最终选定三次样条插值函数作为拟合函数。同时为了能够高效而准确地筛选出测试点，我们选用了遗传算法，最终进行了多次优化来试图找到最优解。

2 拟合或插值方法选取探究

2.1 成本计算

我们从所给的 400 组数据中随机抽取部分数据做出其曲线图，发现大多数数据都有近似的曲线性状。考虑到，我们将先依据这些数据确定一种较好的方案实现对未知器件可以通过取少量的点来实现拟合。并建立如下的评分规则：

- (1) 公式单点定标误差成本按照式（2-1）评估
- (2) 单点测定成本即实施一次单点测定的成本以符号 q 记。本课题指定 $q=12$ 。
- (3) 某一样本的定标成本按照公式（2-2）评估。
- (4) 按照式（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。

依据这样的评分规则研究每一组数据，我们将得到的曲线点与其原数据值比较得到分数，去所有组得分的平均值作为本组解的得分。

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

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

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。

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

2.2 三次样条插值拟合

2.1.1 插值方法的选择

在数值分析中，插值过程可以具体使用线性插值，三次样条插值，立方插值等操作。在曲线插值中最常用的是线性插值法，它是估计两个主干点之间数值的最简单，最易行的方法，

但由于其不能显示连接主干点间的凸状弧线，因此我们决定使用三次样条插值法进行拟合。

2.1.2 三次样条插值拟合的概述

样条插值使用一种特殊分段的分段多项式（称其为“样条”）进行插值的形式。样条插值可以使用低阶多项式样条来实现较小的插值误差，这样就很好得规避了使用高阶多项式所

出现的“龙格”现象。对于 $(n+1)$ 个给定的数据集合 $\{x_i\}$ ，我们可以使用 n 段三次多项式在数据点之间构建一个三次样条。用公式（2-4）表示对函数 f 进行插值的样条函数， $S(x)$ 是分段函数，取值如下：

$$S(x) = \begin{cases} S_0(x), x \in [x_0, x_1] \\ S_1(x), x \in [x_1, x_2] \\ \dots \\ S_{n-1}(x), x \in [x_{n-1}, x_n] \end{cases} \quad (2-4)$$

$S(x)$ 满足以下特性：

- （1）插值特性： $S(x_i) = f(x_i)$
- （2）样条相互连接： $S_{i-1}(x_i) = S_i(x_i), i = 1, 2 \dots n-1$
- （3）两次连续可导： $S'_{i-1}(x_i) = S'_i(x_i)$ 以及 $S''_{i-1}(x_i) = S''_i(x_i), i = 1, \dots, n$

由于对于每个三次多项式函数需要有四个参数如公式 $P = \text{polyfit}(x, y, n)$ （2-5）式中： x, y 为已知数据， n 为拟合多项式的阶次， p 为返回所得多项式的系数向量。所以对于组成 S 的 n 个三次多项式而言，需要 $4n$ 个条件确定这些多项式。

$$y = ax^3 + bx^2 + cx + d \quad (2-6)$$

3 遗传算法

3.1 遗传算法的概述

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。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。

遗传算法是解决搜索问题的一种通用算法，对于各种通用问题都可以使用。搜索算法的共同特征为：

- （1）首先组成一组候选解；
- （2）依据某些适应性条件测算这些候选解的适应度；
- （3）根据适应度保留某些候选解，放弃其他候选解；
- （4）对保留的候选解进行某些操作，生成新的候选解。

在遗传算法中，上述几个特征以一种特殊的方式组合在一起：基于染色体群的并行搜索，带有猜测性质的选择操作、交换操作和突变操作。这种特殊的组合方式将遗传算法与其它搜索算法区别开来。

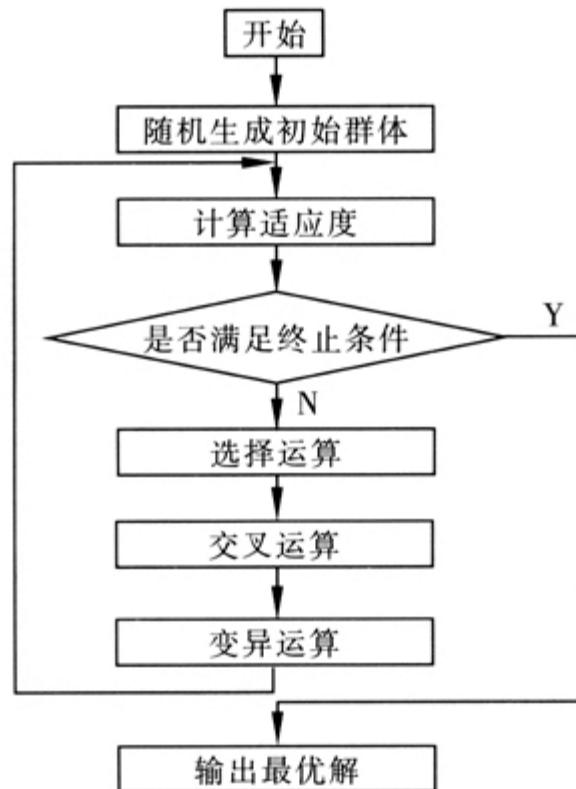


图 2 遗传算法的计算流程

3.2 遗传算法参数设置

为了直观和方便，我们采用了 MATLAB 自带的 Optimization 工具箱实现遗传算法。

我们在进行计算时，选取成本函数作为适应度函数，种群类型为二进制串，种群大小为 51，代表可选取的 51 个测试点，种群数量为 200。规定交叉率为 0.8，变异率为 0.01。算法的终止条件为代数达到 500，或 50 代内适应度变化小于规定值。

其他选项均选用 Optimization 工具箱的默认设置。

3.3 遗传算法计算结果

计算得到的最佳个体为：

0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	1	0						

相当于依次选取 X 为 5.1，5.8，6.9，7.5，8.2，9.2，9.9 的 7 个测定点，这一方案的平均成本为 95.1337。

3.4 遗传算法的不足

通过使用遗传算法计算最优解的过程中，我们小组也发现了许多的遗传算法的不足：

- (1) 遗传算法的计算量非常的庞大，因为计算机进行分段多次拟合的过程耗时比较大，通过我们遗传算法设置的初始参数，初始种群 200 以及运算 500 代使得电脑需要运行成千上万次上述过程，导致总运行时间非常的长。
- (2) 交叉用的是单点法，直接交叉两段基因，所以不利于搜索的全面性，易陷入局部极值。
- (3) 在选择下一代群体时，最佳个体的生存机会将显著增加，最差个体的生存机会将被剥夺，低适值个体淘汰太快容易使算法收敛于局部最优解。

4 结束语

通过对统计推断在模数、数模转换系统中的应用实例的分析,我们得到一种在工程应用中能够减少测量工作量并快速获得类似于本例中 X - Y 函数的一般方法,即通过遗传算法和三次样条插值拟合法,求得某个输入输出系统的特征点分布。对于之后的同样系统特性的测量,我们只需要测量器在特征点的输入输出特性,即可获得该系统的输入输出关系。这在工程应用中可以节约时间以及生产成本。

最后,首先感谢袁焱老师的悉心指导以及助教老师的辛勤工作,其次很开心当面聆听老师的指导意见,还要感谢一直辛勤付出的队友。

5 参考文献

- [1] 袁焱. 统计推断在数模转换系统中的应用课程讲义[EB/OL]. <ftp://202.120.39.248>.
- [2] 维基百科. 遗传算法
[J/OL]. <http://zh.wikipedia.org/wiki/%E9%81%97%E4%BC%A0%E7%AE%97%E6%B3%95>.
- [3] 百度文库. Matlab 遗传算法工具箱及其应用.
http://wenku.baidu.com/link?url=2FN2TqDeh0e6VoNThkIic809Ldmvk5NF8_psLPT-61ld4yiFhZ02oy3NCH7eSzIqjs0l0gKlhYiMazslo8_8UNSOBFnufFxe4FP1xjlAtFC&qq-pf-to=pcqq.c2c.

附录:

fitness.m

```
function f=fitness(gene)
if sum(gene)>15 || sum(gene)<2
    f=+Inf;
else
dataform=importdata('dataform.mat');
gene=logical(gene);
s=0;
for i=2:2:800
    x=5:0.1:10;
    xt=x(gene)';
    y=dataform(i,:);
    yt=y(gene)';
    yi=interp1(xt,yt,x,'spline');
    diff=abs(y-yi);
    for j=1:51
        if diff(j)<=0.4
            s=s+0;
        elseif diff(j)<=0.6
            s=s+0.1;
        elseif diff(j)<=0.8
            s=s+0.7;
        elseif diff(j)<=1
            s=s+0.9;
        elseif diff(j)<=2
            s=s+1.5;
        elseif diff(j)<=3
            s=s+6;
        elseif diff(j)<=5
            s=s+12;
        else
            s=s+25;
        end
    end
end
f=s/400+sum(gene)*12;
end
end
```

test_your_answer.m

%%%%%%%% 答案检验程序 2015-11-04 %%%%%%%%%

my_answer=[2,9,20,26,33,43,50]; % 选点组合

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

 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(nsampl e,1)*my_answer_n;  
cost=sum(si)/nsampl e;
```

```
% 显示结果
```

```
fprintf('\n经计算，你的答案对应的总体成本为%5.2f\n',cost);
```

```
mycurvefitting.m
```

```
function y1 = mycurvefitting( x_premea,y0_premea )
```

```
x=[5.0:0.1:10.0];
```

```
% 采用三次样条插值
```

```
y1=interp1(x_premea,y0_premea,x,'spline');
```

```
end
```