

统计推断在数模转换系统中的应用

组号 61 姓名：沈扬 学号：5120309519，
姓名：张璟聪 学号 5120309521

摘要：本报告主要展示了统计推断在模数、数模转换系统中的应用，以为某种电子产品模块的传感特性校准设计方案为研究实例，研究和分析 469 组电压信号 X 和被测物理量 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 建立电压 X 和被测物理量 Y 的关系曲线模型，最终实现以少量数据反映整体系统特性的效果，从而有效降低工程设计上的成本，提高效率。

关键词：样本，特征点，曲线拟合，样条差值法，优化采样，遗传算法

ABSTRACT: This report mainly shows the application of statistical inference in the module, d/a conversion system, thought some electronic products module of sensor calibration feature design as the research example, research and analysis of 469 groups of voltage signal the relationship between the measured quantities X and Y , using certain mathematical statistics method, a feature point selection, algorithm research, fitting, comparison and a series of process with the aid of MATLAB to establish the relationship between the voltage measured quantities X and Y curve model, finally realizes with a small amount of data reflect the effect of the overall system performance, thus effectively reducing the cost of the engineering design, improve efficiency.

Key words: sample, feature point, curve fitting, Spline difference method, optimized sampling, GA

1 引言

在生产和研究中，对物理现象的研究可以通过确定其输入输出的关系来确定其性质，这就有必要提出一种解决方案来确定这二者的关系。但是实际研究生产中发现，我们很难找到一个能够精准描述二者关系的函数，这就要求我们运用实际测得的数据，通过曲线拟合来找到一个最符合输入与输出关系的函数。为了保证准确找到最优解，我们就需要运用统计推断的知识找到合适的特征点，对这些特征点进行拟合，计算得到的曲线与实际输出的残差，残差最小的解围最优解。本次统计推断的报告是通过研究电子产品输入的电压信号 X 与被监测的物理量 Y 的关系，分析大量的数据得出最优解，来说明统计推断在获得输入输出关系时的作用。

1.1 任务概述

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。

1.2 设计思路

基本任务中主要有两个方面的问题需要考虑：

一是如何选取 7 个特征点，虽然说暴力穷举法从理论上来说是可行的，但对于我们所使用的个人电脑来说运行所需要的时间过于长，并且从 51 个点中枚举所有 7 个特征点的组合来尝试拟合显然没有必要，因此使用适当的组合优化方法是必要的。

二是用何种方法还原曲线的表达式，这里采用三次多项式拟合法。

1.3 具体模型

为了对本课题展开有效讨论，需建立一个数学模型，对问题的某些方面进行必要的描述和限定。

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）。

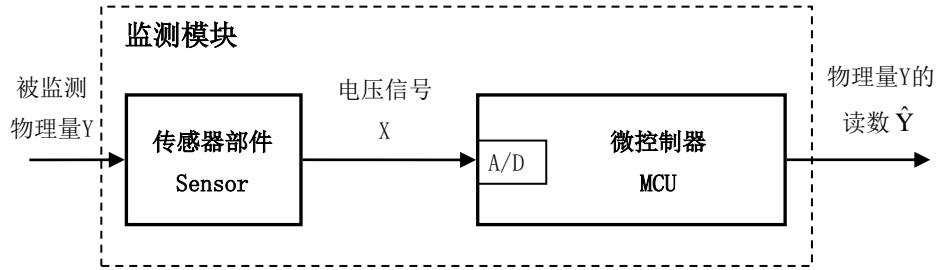


图 1 监测模块组成框图

1.4 评价函数

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

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

单点定标误差的成本按式（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 取点方式

2.1 取点数目的分析

就该系统而言,测定 51 组数据后,系统特性已经被充分定标,若对每个样本都进行 51 组数据的完整测定,既费时间又耗成本,没有直接的工程实用意义。我们需要得到一种根据较少的点数就能得到较好的 Y-X 函数关系的一种实现方法。

经过后期多次拟合尝试的结果确定,我们在这里认为取 7 个点为能达到预期效果,且计算量合适。

2.2 取点区间的划分

为了大致确定这 7 个点应该存在的范围,故需要对 Y-X 特性做大致的划分,以便使 7 个点的选取不至于过于集中,使得 7 个点在相应的范围内变动,这样也可以大大减少接下来的计算量。

区间划分应该具有普适性,即能够对 469 组实验数据同时具有划分的意义。由于 X 的坐标一样,故以 X 为横坐标,绘制 Y-X 图,并通过对 Y 求二阶差分来判断电压 X 划分点。

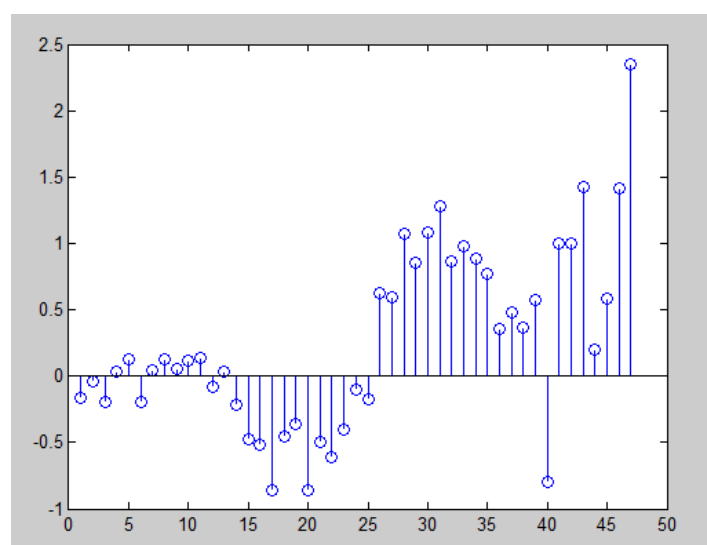


图 2 Y-X 的二阶差分离散图

由图 2 可以看出,图像大致分为三段,即 1-12 点为一段,13-26 点为一段,26-51 点为一段。

由此可确定本次实验 7 个点取点方法:第一段 2 个点,第二段 2 个点,第三段 3 个点。

3 数学模型

根据曲线的大致形状的划分方法,建立三次多项式拟合和三次样条插值法的数学模型。

3.1 三次多项式拟合

此方法为拟合的常用简单方法,即设一在区间 $[a, b]$ 上的 n 阶多项式函数

错误!未找到引用源。

(3)

其中 $a=x_0 < x_1 < \dots < x_n=b$ 是数据点,则它为区间 $[a, b]$ 上对数据点拟合的多项式。由定义可看出,它有 $n+1$ 个待定系数。确定这些系数 a_k 使

错误!未找到引用源。 (4)

最小。这时就得到误差平方拟合多项式。可以把它认为是推断函数。同样,我们利用 MATLAB 中已有的函数

错误!未找到引用源。 (5)

进行多项式拟合。其中输入参数 x_0, y_0 为要拟合的数据, m 为拟合多项式的次数, 输出参数 a 为拟合多项式系数

错误!未找到引用源。 (6)

系数 $a=[a_m, L, a_1, a_0]$ 。多项式在 x 处的值 y 可用 MATLAB 中的 $y=polyval(a,x)$ 函数进行计算。

多项式拟合的次数在一定范围内越高, 残差、方差等参数越小, 拟合曲线与实际测量点的相关性越高, 拟合程度越好。但次数的增高导致算法运行时间的延长, 效率的降低, 而且当次数超过一定范围时, 甚至适得其反。例如用 5, 6 次函数拟合, 运行时间是很难让人接受的, 所得结果误差反而越来越大, 可见不一定次数越高, 拟合曲线就越接近实际曲线。反复测试后, 3 次曲线拟合的效果最好。

3.2 三次样条插值法

假设在 $[a,b]$ 上测量有 $n-1$ 个点, 在 $[a,b]$ 上划分。 $\Delta: a=x_0 < x_1 < \dots < x_{n-1} < x_n=b$, 则三次样条插值法所得的函数 $S(x)$ 具有如下特征^[1]:

- (1) 在每个小区间 $[x_j, x_{j+1}]$ 上是三次多项式。
- (2) 在每个节点 $S(x_j)=y_j$ 。
- (3) $S(x)$ 在 $[a, b]$ 上二阶导数连续。

因为 $S(x)$ 在 $[a, b]$ 上二阶导数连续, 所以在每个节点, 由连续性得 $S(x_j-0)=S(x_j+0)$; $S'(x_j-0)=S'(x_j+0)$; $S''(x_j-0)=S''(x_j+0)$ 。共 $3n-3$ 个条件; 除此之外, 由插值可得 $n+1$ 个条件, 仍需两个, 这两个条件有边值得出:

- (1) 给定断点 x_0, x_n 处的一阶导数值
- (2) 给定断点 x_0, x_n 处的二阶导数值 (特别的 $S''(x_0+0)=0$, $S''(x_n+0)=0$ 称为自然边值条件)
- (3) 周期性便捷条件: 即以 $b-a$ 为周期, $S''(x_0+0)=S''(x_n+0)$, $S'(x_0+0)=S'(x_n+0)$ 。

三次样条插值函数在 MATLAB 中可以使用 `spline` 函数实现。

3.3 运行结果

3.3.1 三次多项式拟合

取点: 3, 10, 20, 26, 35, 43, 49

MIN = 95.12。具体 MATLAB 代码见附件[1]。

3.3.2 三次样条插值

取点: 3, 8, 16, 25, 35, 44, 50

MIN = 96.94。具体 MATLAB 代码见附件[2]

3.4 三次样条插值法相比于三次多项式拟合优点

三次样条插值法相比于七点法对所测量得到的信息的应用足够充分, 最后所得的曲线必过每个测量点, 在每两个测量点之间都有自己的三次多项式。七点法最后所得的式子不一定过测量点。即用三次样条插值法所得的拟合方程, 更加贴近真实的测量结果, 同时也没有丧失多阶导数连续性。不过时间长效率低是一个不足。

4 遗传算法 (GA)

4.1 遗传算法简介

遗传算法就是模拟进化论的优胜劣汰的规则，并加有概率性质的随机遗传的算法。算法中产生个体采用产生 1 至 51 的 7 个随机数的方法，而从父代向子代的遗传主要分三种方式：

- (1) 选择 (selection)，选择父代中优秀的个体直接传递给下一代，即保留父代。
- (2) 交叉 (crossover)，以两个父代的基因为基础，进行交叉重组，生成下一代，好的基因会有更大的概率传给子辈。
- (3) 变异 (mutation)，每个父代个体都有一定的概率发生随机的基因突变。

评价基因好坏的依据就是适应度。在遗传算法的步骤中起着突出作用的其实是变异。正因为有了变异，算法才存在在全局范围内寻找到最优解的可能性。但是遗传算法需要通过反复试验，才有可能给出最优解。

4.2 流程图

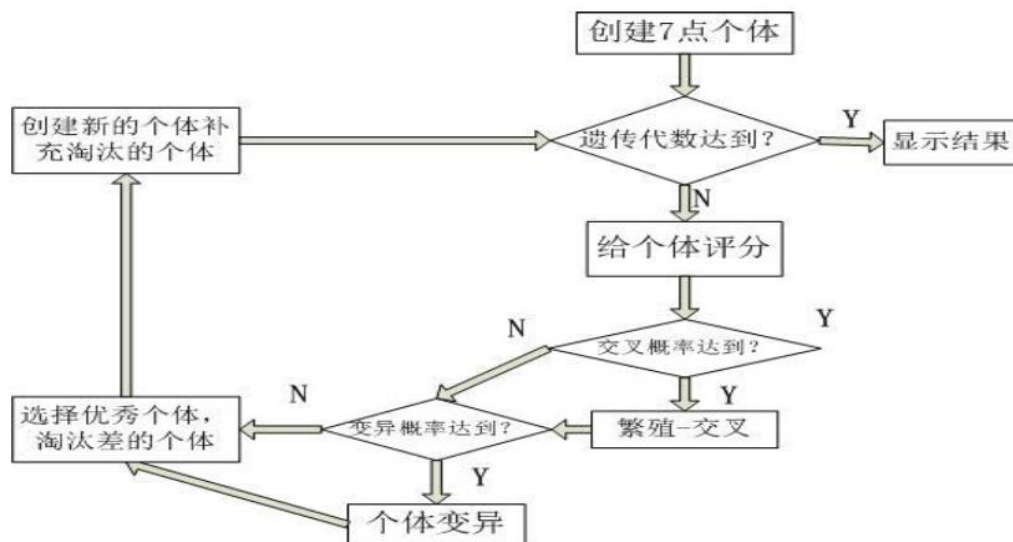


图 3 遗传算法流程图

4.3 实现过程

- (1) 编码：采用二进制编码，即用二进制数构成的符号串来表示一个个体。
- (2) 解码：编码后的个体构成的种群必须经过解码，以转换成原问题空间的决策变量构成的种群，方能计算相应的适应值。
- (3) 选择：选择过程是利用解码后求得的各个体适应值大小，淘汰一些较差的个体而选出一些比较优良的个体，以进行下一步的交叉和变异操作。
- (4) 交叉：采用单点交叉的方法来实现交叉算子，即按选择概率 PC 在两两配对的个体编码串中随机设置一个交叉点，然后在该点相互交换两个配对个体的部分基因，从而形成两个新的个体。
- (5) 变异：对于二进制的基因串而言，变异操作就是按照变异概率随机选择变异点，在变异点处将其位取反即可。

4.4 参数设置及运行结果

初始种群: 250
 交叉概率: 0.6
 变异概率: 0.08
 遗传代数: 30
 取点: 4, 12, 18, 29, 36, 47, 50
 MIN =94.58

4.5 方法评价

这种简单的遗传算法结果较精确可是运行时间过长。原因如下:

- (1) 所用数据库庞大, 有 469 组之多, 程序多次执行求 469 组平均分, 耗时大。
- (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。
- (3) 变异采用的是单点变异, 效率较低。
- (4) 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。

5 总结

本次使用三次多项式拟合, 三次样条插值拟合分别对取 7 个点进行了模拟, 三次样条插值结果优于三次多项式拟合, 但所花时间也更多; 对于简单的“启发式算法”遗传算法 (GA), 刚开始不能很好地明白具体意义, 咨询老师及查询各种资料后才较为艰难的完成了编程得出了结论。

这门课让我们从一开始的不知所措到后来的慢慢了解课题, 讨论, 开始编程到最后的启发式算法, 让我更好地开发了自己的思路, 自学了不少知识, 对 MATLAB 了解也增多了。

最后感谢老师对我们问题的解答和帮助!

6 参考文献

- [1] 上海交大电子工程系. 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] <http://www.cnblogs.com/biaoyu/archive/2011/12/03/2274604.html>

附

(1) 二阶差分代码

function chafen()	Y1=sum(Y(:,i+1))-sum(Y(:,i));
-------------------	-------------------------------

<pre> global data; score=0; for k=1:2:1289%%改 f=spline(data(k,q),data(k+1,q),data(k,1:53)); p=0; for n=2:1:50 f=spline(data(k,q),data(k+1,q),data(k,n)); if (abs(f(n)-data(k+1,n)))<0.01 p=p+2; else if (abs(f(n)-data(k+1,n)))<0.03 p=p+1.8; else if (abs(f(n)-data(k+1,n)))<0.05 </pre>	<pre> p=p+0.4; else if (abs(f(n)-data(k+1,n)))<0.15 p=p+0; else p=p-20; end end p=max(p-2,0); score=score+p; end score=score/469; end </pre>
--	---

(4) 遗传算法

<pre> function initilize(350, 28) global pop; for i=1:350 for j=1:28 pop(i, j) = round(rand); end end clear i; clear j; function fitness(350, 28) global fitness_value; global pop; global G; for i=1:350 fitness_value(i) = 0.; end for i=1:350 for j=1:28 if pop(i, j) == 1 fitness_value(i) = fitness_value(i)+2^(j-1); end end fitness_value(i) = -1+fitness_value(i)*(3.-(-1.))/(2^28-1); fitness_value(i) = -(fitness_value(i)-1).^2+4; </pre>	<pre> global best_fitness; global best_individual; global best_generation; global pop; global G; for i=1:350 fitness_table(i) = 0.; end min = 1; temp = 1; templ(28)=0; for i=1:350 min = i; for j = i+1:350 if fitness_value(j)<fitness_value(min); min = j; end end if min~=i temp = fitness_value(i); fitness_value(i) = fitness_value(min); fitness_value(min) = temp; for k = 1:28 templ(k) = pop(i, k); pop(i, k) = pop(min, k); </pre>
--	--

<pre> end clear i; clear j; function rank(350, 28) global fitness_value; global fitness_table; global fitness_avg; for i=1:350 if i==1 fitness_table(i) = fitness_table(i) + fitness_value(i); else fitness_table(i) = fitness_table(i-1) + fitness_value(i); end end fitness_table fitness_avg(G) = fitness_table(350)/350; if fitness_value(350) > best_fitness best_fitness = fitness_value(350); best_generation = G; for j=1:28 best_individual(j) = pop(350, j); end end clear i; clear j; clear k; clear min; clear temp; clear templ; function selection(350, 28, elitism) global pop; global fitness_table; </pre>	<pre> pop(min, k) = templ(k); end end end if r > fitness_table(mid) first = mid; elseif r < fitness_table(mid) last = mid; else idx = mid; break; end mid = round((last+first)/2); if (last - first) == 1 idx = last; break; end for j=1:28 pop_new(i, j)=pop(idx, j); end end if elitism p = 350-1; else p = 350; end for i=1:p for j=1:28 pop(i, j) = pop_new(i, j); end end clear i; clear j; clear pop_new; clear first; </pre>
---	--

<pre> for i=1:350 r = rand * fitness_table(350); first = 1; last = 350; mid = round((last+first)/2); idx = -1; while (first <= last) && (idx == -1) function crossover(350, 28, 1) global pop; for i=1:2:350 if(rand < 1) cross_pos = round(rand * 28); if or (cross_pos == 0, cross_pos == 1) continue; end for j=cross_pos:28 temp = pop(i, j); pop(i, j) = pop(i+1, j); pop(i+1, j) = temp; end end end end clear i; clear j; clear temp; clear cross_pos; function mutation(350, 28, mutate_rate) global pop; for i=1:350 if rand < mutate_rate mutate_pos = round(rand*28); if mutate_pos == 0 continue; end pop(i,mutate_pos) = 1 - pop(i, mutate_pos); </pre>	<pre> clear last; clear idx; clear mid; plot(x,y) function [m,n,p,q] = GeneticAlgorithm(350, 28, 30, 1, mutate_rate, 1) global G ; global fitness_value; global best_fitness; global fitness_avg; global best_individual; global best_generation; fitness_avg = zeros(30,1); disp "hhee" fitness_value(350) = 0.; best_fitness = 0.; best_generation = 0; initilize(350, 28); for G=1:30 fitness(350, 28); rank(350, 28); selection(350, 28, 1); crossover(350,28,1); mutation(350, 28, 0.08); end plotGA(30); m = best_individual; n = best_fitness; p = best_generation; q = 0.; for j=1:51 if best_individual(j) == 1 q = q+2^(j-1); end end q = -1+q*(3.-(-1.))/(2^28-1); </pre>
--	---

<pre>end end clear i; clear mutate_pos; function plotGA(30) global fitness_avg; x = 1:1:30; y = fitness_avg;</pre>	<pre>clear i; clear j;</pre>
---	------------------------------