

统计推断在数模转换系统中的应用

组号 52 姓名 陈琦 学号 5120809040, 姓名 刘硕 学号 5120309340

摘要: 本文以某型产品内部的一个检测模块为研究对象, 为该产品寻求校准工序的优化方案。 X 为某个特性值, Y 为被监测的相应物理量。在确保测量精度的前提下, 运用统计方法结合 Matlab 对样本中 $Y-X$ 数据关系进行分析, 选取函数模型对其进行拟合, 并在每个样品中找出 N 个点从而确定拟合函数的参数, 要求此方法确定的函数对所有样品的平均测量成本最低。本文讨论了不同拟合方式的效果, 并用遗传算法找出了最优的取点组合。我们还利用假设检验对结果进行了数值分析。

关键词: 后项差分, 归一化, 三次样条插值法, 多项式拟合, 分段拟合, 遗传算法, matlab, 假设检验, Lilliefors 检验

Statistic Application On The AD-DA Converting System

ABSTRACT: This article is to introduce the statistical inference to analog-to-digital conversion and then to simplify the progress of sampling in engineering practice. To achieve the goal, we use several ways such as polynomial fitting, cubic spline interpolation and segmentation fitting to fit the function of sample points. Genetic Algorithm is employed to locate the optimal combination of sampling points. Then we use Lilliefors Testing to analyze our results.

Key Words: Consequent differential, uniformization, cubic spline interpolation, polynomial fitting, segmentation fitting, Genetic Algorithm, matlab, Lilliefors Testing

1. 课题概述

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。

2. 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 2-1。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)。

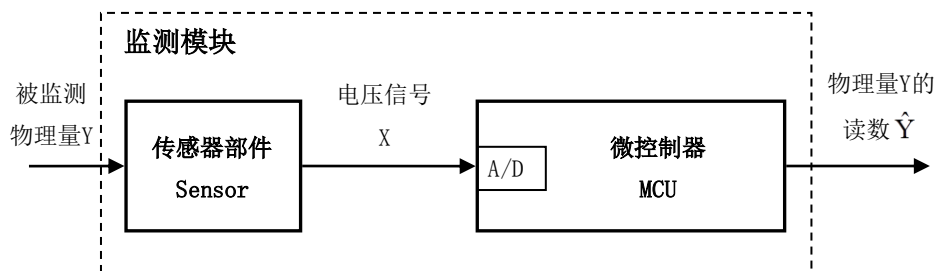


图 2-1 监测模块组成框图

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。[1]

考虑实际工程中该监测模块的应用需求，同时为便于在本课题中开展讨论，我们将问题限于 x 为离散取值的情况，规定

$$X \in \{x_1, x_2, x_3, \dots, x_{50}, x_{51}\} = \{5.0, 5.1, 5.2, \dots, 9.9, 10.0\}$$

相应的 Y 估测值记为 $\hat{y}_i = f(x_i)$ ， Y 实测值记为 y_i ， $i = 1, 2, 3, \dots, 50, 51$ 。

2.1 传感部件特性

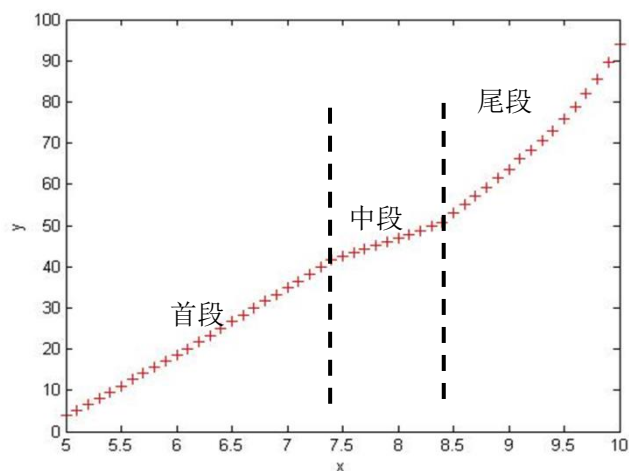


图 2-2 传感特性图示

一个传感部件个体的输入输出特性大致如图 2-2 所示，有以下主要特征：

- (1) Y 取值随 x 取值的增大而单调递增；
- (2) x 取值在 $[5.0, 10.0]$ 区间内， Y 取值在 $[0, 100]$ 区间内；
- (3) 不同个体的特性曲线形态相似但两两相异；
- (4) 特性曲线按斜率变化大致可以区分为首段、中段、尾段三部分，中段的平均斜率小于首段和尾段；
- (5) 首段、中段、尾段单独都不是完全线性的，且不同个体的弯曲形态有随机性差异；
- (6) 不同个体的中段起点位置、终点位置有随机性差异。

为进一步说明情况，图 2-3 对比展示了四个不同样品个体的特性曲线图示。

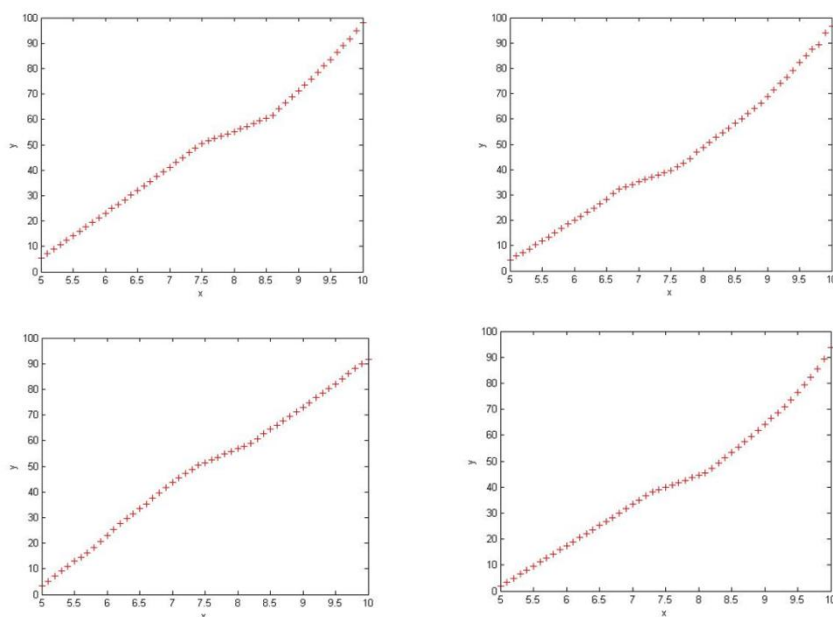


图 2-3 四个不同样本个体特性图示对比

2.2 标准样本数据库

前期已经通过试验性小批量生产，制造了一批传感部件样品，并通过实验测定了每个样品的特性数值。这可以作为本课题的统计学研究样本。数据被绘制成表格，称为本课题的“标准样本数据库”。

该表格以 csv 格式制作为电子文件。表格中奇数行存放的取值，偶数行存放对应的取值。第 $2i-1$ 行存放第 i 个样本的 X 数值，第 $2i$ 行相应列存放对应的实测 Y 数值。

2.3 成本计算

为评估和比较不同的校准方案，特制定以下成本计算规则。

(1) 单点定标误差成本

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

单点定标误差的成本按式 (1-1) 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) 单点测定成本

实施一次单点测定的成本以符号 q 记。本课题指定 $q=12$ 。

(3) 某一样本个体的定标成本

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

对样本 i 总的定标成本按式（2-2）计算，式中 n_i 表示对该样本个体定标过程中的单点测定次数。

（4）校准方案总体成本

按式（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。

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

总体成本较低的校准方案，认定为较优方案。

3. 拟合方式的选取

3.1 不分段多项式拟合

由泰勒展开式理论可知，每一个函数都可以展开为多项式的级数，所以不管拟合的表达式为什么函数，均可以用多项式的组合来表示。

设：

$$Y_i = \sum_{n=0}^N a_n X_i^n + \epsilon_i \quad \text{其中 } i=1, 2, \dots, 51 \tag{3-1}$$

标准差为：

$$s = \sqrt{\frac{\sum_{i=1}^{51} \epsilon_i^2}{51}} = \sqrt{\frac{\sum_{i=1}^{51} (Y_i - (\sum_{n=0}^N a_n X_i^n))^2}{51}} \tag{3-2}$$

欲使 s 达到最小，等价于求 $s(a_1, a_2, \dots, a_N)$ 的最小值点，利用 Origin 软件直接对数据进行多项式拟合。选取前 10 组数据不同次数多项式拟合的计算结果见表 3-1。

表 3-1 不分段拟合对于不同次数的比较

	2 次多项式	3 次多项式	4 次多项式	5 次多项式	6 次多项式	7 次多项式	8 次多项式
S 均值	1. 41936	1. 45337	0. 94236	0. 88446	0. 53165	0. 58676	0. 59322

以上结果表明，拟合的阶数越高，精度也越高，但因为“龙格现象”的存在[2]，即阶数越高而插值结果越偏离原函数，7 次时 s 又开始回升。故本文只尝试 3~6 阶多项式拟合。

3.2 分段多项式拟合

3.1.1 分段依据

直观上看图像斜率变化也可明显看出曲线分为 3 段。

因为数据点是离散的，无法计算斜率，故计算后项差分：

$$\Delta_i=Y_{i+1}-Y_i,i=1,3,\cdots,50$$

(3-3)

通过 EXCEL 计算前 100 数据组的后项差分，发现其绝对值并无明显的统计规律，所以又找出每一组后项差分的最大值 Δ_{\max} ，对各个点的差分进行归一化：

$$\Delta_i'=\Delta_i/\Delta_{\max}$$

(3-4)

对归一化后的后项差分作直方图，发现其分布规律大致相同，见图 3-1. 中间出现“断层”，可以推断分为 3 段。且中间段后项差分基本相等，可以推断中间一段呈线性。

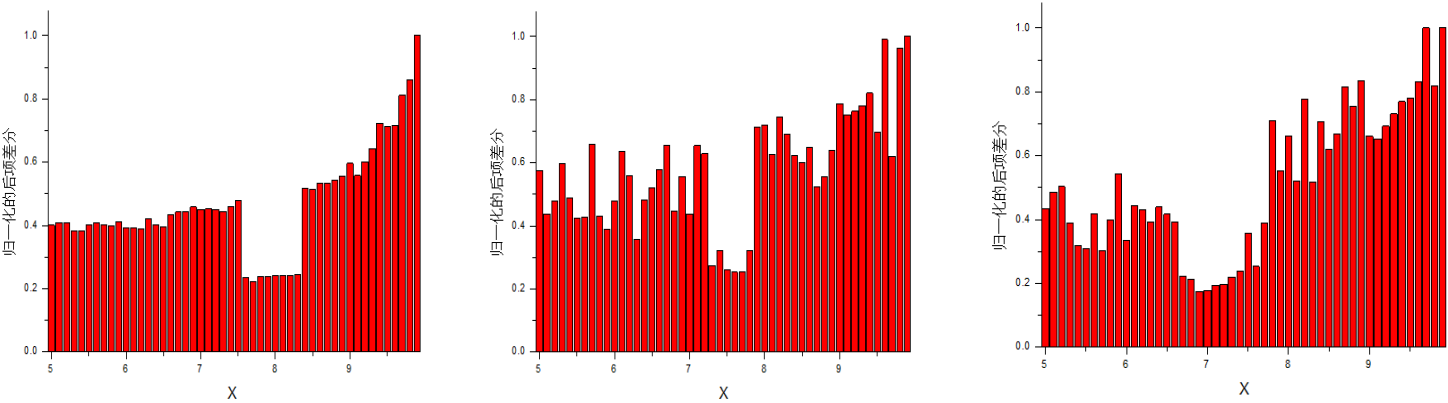


图 3-1 3 组数据归一化后的后项差分分布

3.1.2 分段区间的确定

经统计 100 组数据发现，归一化的后项差分第一段和第三段的值均大于 0.4，而中间段在 0.2 左右，故采取 0.3 为分界点，记 100 组中各个点归一化的后项差分小于 0.3 的概率 p，人为的限定 p>0.6 的点落在线性区间内，从而筛选出 6 个点。（原本预期这个概率应该在 0.9 以上，可是实际 p 都不会很大，原因可以从图 3-1 上看出，对比不同组，可以发现虽然每一组都明确地分了三段，但中间段的起止点以及宽度都是因组而异的，所以给所有组统一分区间是不精确的，可行的方法是在每一组拟合的时候都以同样的方法判断转折点，但是程序较复杂这里就不尝试了）。

表 3-1 中段点的选取

X	7.2	7.3	7.4	7.5	7.6	7.7
p	0.6	0.62	0.64	0.72	0.72	0.64

从而划分区间[5,7.1],[7.2,7.7],[7.8,10]。对头尾两端采取多项式拟合，对中间段线性拟合。取前十组数据，分区间进行多项式拟合，分析标准差 s 的均值得

表 3-2 首尾段的拟合

S 均值	线性	2 次多项式	3 次多项式	4 次多项式
[5, 7.1]	0.26422	0.23014	0.20544	0.21176
[7.2, 7.7]	0.13977	0.22555		

[7.8, 10]	0.90755	0.47867	0.28885	0.24874
-----------	---------	---------	---------	---------

第一段次数增加时标准差下降不大，考虑到取点个数带来成本，选择线性拟合，映射区间[1,22]；

第二段线性拟合，映射区间[23,28]；

第三段采取三次多项式拟合[29, 51]。

3.1.3 三次样条插值法

使用 Matlab 自带 interp1 函数

原理：

假设在[aa,bb]上测量有n-1个点，在[aa,bb]上划分。 $\Delta: =xx_0 < xx_1 < \cdots < xx_{nn-1} < xx_{nn} = bb$ ，则三次样条插值法所得的函数S(x)具有如下特征：

(1) 在每个小区间[xx_{jj},xx_{jj+1}]上是三次多项式。

(2) 在每个节点SS(xx_{jj})=yy_{jj}

(3) SS(xx)在[aa,bb]二阶导数连续。

因为SS(xx)在[aa,bb]上二阶导数连续，所以在每个节点，由连续性得S_{xj+0}=S_{xj-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xx₀,xx_{nn}处的一阶导数值

(2) 给定断点xx₀,xx_{nn}处的二阶导数值（特别的S''(x₀₊₀)=0, S''(x_{nn+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''(x₀₋₀)=S''(x_{n+0})，S'(x₀₋₀)=S'(x_{n+0})。 [3]

3.1.4 其他增长函数模型

表 3-3 几个特殊函数的模型

函数模型	Boltzman 函数	Logistic	Dhyperbl
	$y = \frac{A_1 - A_2}{1 + e^{(x-x_0)/\Delta k}} + A_2$	$y = \frac{A_1 - A_2}{1 + (x/x_0)^p} + A_2$	$y = \frac{p_1 x}{p_2 + x} + \frac{p_3 x}{p_4 + x} + p_5 x$
方差均值	3.47098	1.45337	0.1244

方差均值为前 10 组数据带入 Origin 软件算得的平均值。由此可以看出 Dhyperbl 函数的拟合效果很好，甚至优于之前讨论的多项式拟合。

若对 Dhyperbl 函数进行拟合，至少需要 6 个点确定参数，这是一个 5 元 3 次方程，得出来的解还可能不唯一，也有可能是复数。目前我还没想到有效的求解的方法（遗传算法等优化算法可以求近似解，那这个课题的求解就变成恶性循环了）。

若想在有解的前提下得到唯一稳定的解，可对其进行展开：

$$(p_2 p_4 + (p_2 + p_4)x + x^2)y = p_1 p_4 x + p_1 x^2 + p_2 p_3 x + p_3 x^2 + p_5 p_2 p_4 x + p_5 p_2 x^2 + p_5 p_4 x^2 + p_5 x^3$$

可将与 p 有关的每一项看做方程的解，则原方程化为 10 元一次方程，可得稳定的解，但需

要 11 个 x 的值，此时成本已经大于 132。所以即使此函数参数能够确定，即使它的精确度成本很小，取点成本也很高，在工程上不具备应用价值，此处也不继续深究了。

4. 特征点的选取

要在每组第 2-50 个数据中选出尽可能少的点，比如从 5 个点开始，有

$C_{50}^5 + C_{50}^6 + C_{50}^7 + \dots$ 种选择，若用穷举法，是个 NP-hard 问题。为此，引入遗传算法，

对以上拟合方式分别计算结果。

4.1 遗传算法的流程框图



4.2 遗传算法的具体实现

- 1、初始种群的产生: `randperm(49,n)+ones(1,n)` 产生 n 个 2-50 之间的不同基因
- 2、成本计算函数: 之前自己用判断语句写了一个, 后来发现老师的测试文件效率高, 就用老师的了。选择压函数: 直接用成本作为选择条件, 成本越低, 被选择保留。
- 3、单点突变: 当产生的随机数小于突变概率时发生突变, 随机选取一个基因发生突变, 有变步长, 当收敛时变步长也相应减小。当发生不利突变或变异不合法时, 保留亲代, 去除子代。
- 4、交配: 先按成本排序, 前一半为劣势个体, 后一半优势个体之间相邻的交配, 优势与劣势个体对应交配。变异不合法或成本没有下降时, 保留亲代, 舍去子代。[4]

5. 结果表示

在 matlab 调用 MyGA(200,n,200,0.5,0.01),表示种群规模为 200，基因个数为 n，经历 200 代，交叉概率为 0.5，变异概率为 0.01。改变 n 的值可得不同取点个数下的最大值。每运行一次耗时 2000 秒左右。对于不分段拟合，只有成本计算函数中的拟合函数有差异，分段拟合函数可以看成不同区间下的多项式拟合，仅需改变初始种群，excel 读入数据范围，以及突变的范围即可。

不分段拟合时，三段之间是否采用公共点进行拟合，会对结果造成很大影响，因为少用一个点，成本会降低 12。再尝试了不使用公共点拟合之后，考虑到分段之时分界点本来也不明确，所以将线性部分的最后一个点 28 用作第二段和第三段的公共点，至于为什么不继续将第一段和第二段也采用公共点，因为第一段和第二段都是线性拟合，而分界点又不是很精确，这样造成的精确度成本可能很大。

表 3-2 不同拟合方式下的最优解

拟合方式	特征点个数	特征点组合	成本
不分段 3 次	5	4, 18, 29, 41, 49	113. 0371
不分段 4 次	6	2, 10, 21, 30, 40, 49	104. 7404
不分段 5 次		10 个点以下 matlab 警告	
不分段 6 次		警告	
分段 1+1+3	8	5, 19, 24, 27, 30, 31, 50, 44	104. 4616
分段 1+1+3（固定采用 28 为采样点）	7	5, 19, 24, 28, 35, 44, 50	93. 2313
三次样条插值	6	3, 12, 22, 31, 43, 50	92. 8774

6. 结果分析

比较得出使用三次样条插值得到的结果最理想。

不分段拟合效果不佳的原因在讨论分段拟合时已经揭晓了，每组数据的归一化后项差分结果说明曲线确实是分段的，而且经过不同阶次的分段拟合标准差分析可知，曲线的第一段和第二段可以认为是线性的，而第三段用 3 次拟合效果已经很令人满意，说明问题不在于拟合的阶数，而在于要分段，而且阶数太高会产生“龙格现象”降低精度。

至于分段拟合，虽然更符合曲线的真实走势，但是由于每一组的中间段的起止点和长度都是有差异的，所以无法用统一的规律为所有的组划分区间，区间划分的偏差会提高精确度成本。另外划分区间的固有缺陷就是需要的点个数多，以 1+1+3 方式拟合就需要 2+2+4=8 个点，令一个点为公共点还是需要 7 个点，所以成本还是很高。

而三次样条插值是一个比较“中庸”的方法，它既有分段的因素，又不需要像真正分段那样需要那么多点。

7. 对不同拟合方式得到的结果进行数值分析

Lilliefors 检验:

Lilliefors 检验是改进 K-S 检验并用于一般的正态性检验，原假设 H_0 : 总体服从正态分布 $N(\mu, \sigma^2)$ ，其中 μ, σ^2 由样本均值和方差估计。该检验的 MATLAB 命令 `lillietest`，调用格式 `[H] = lillietest(X,alpha)` 显著性水平 α 在 0.01 和 0.2 之间，缺省时为 0.05。H 为测试结果，若 $H=0$ ，则无法拒绝 X 是服从正态分布的；若 $H=1$ ，则可以否定 X 服从正态分布。[5]

将老师给的 `test_ur_answer` 稍加改动即可对以上结果进行误差分布的正态性进行检验，X 为每一组的误差列向量，显著性水平取默认值 $\alpha = 0.05$ 进行检验。

表 7-1 各拟合方式误差的正态性检验

拟合方式	不分段 3 次拟合	不分段 4 次拟合	三次样条插值
$H = \sum_{i=1}^{469} H_i$	115	116	212
正态分布的比率 $(1 - \frac{H}{469})$	75.5%	75.3%	54.8%

三次样条插值法产生的误差分布正态性并没有那么强，可能原因是每组 51 个数据太少，或者三次样条插值方法本身决定的。这在工程学上的意义可以理解为：因为偶然误差呈正态性，系统误差非正态性，所以在多项式拟合中偶然误差为主，可以采用多次测量取平均的方法减少误差，而三次样条插值中系统误差为主，主要是优越传感器元件噪声的原因造成的，减少误差的办法就是提高其物理性能。

8. 总结

在本文探讨的方法下，最优点组合为 (3, 12, 22, 31, 43, 50)，即 (5.2, 6.1, 7.1, 8.0, 9.2, 9.9)，拟合方式为三次样条插值，成本为 92.8774。

9. 鸣谢

我是选课最后一天才选上这门课的，在此之前一直以为自己选不上这门课，第一节课时没能参加，原本以为由于缺了第一节没能真正理解这个问题的工程学意义，这门课对我来说困难。好在袁老师每次上课前也都反复强调实际意义，而且还有其他组的同学耐心的给我解释课上的内容。感谢李老师在面谈当天给了我不少具体的建设性的意见，比如我的代码风格问题，没有充分利用 `matlab` 矩阵计算的优势使得运算效率低下。袁老师给的测试程序也很有用，可以直接用来作为成本计算函数，稍加改动可以对结果进行假设检验。这门课虽然花费不少精力，但也让我深刻体会到了数学方法在工程学上的运用，不仅仅在于它能简化实际问题，还在于实际问题中要考虑更多的因素，数学问题中的最优解并不一定是实际问题中的最优解。这门课也让我对寒假的数学建模大赛充满了期待。在此感谢给我帮助的老师 and 同

学们!

10. 参考文献

- [1] 上海交大电子工程系 袁焱 统计推断在数模转换系统中的应用课程讲义
- [2] 维基百科. 样条插值 - 维基百科, 自由的百科全书 [M/OL].
<http://zh.wikipedia.org/wiki/龙格现象>
- [3] 陈文略 王子羊 华中师范大学学报(自然科学版) [J]. 三次样条插值在工程拟合中的应用, 2004(12):38-4.
- [4] Andries.P.Engelbrecht 著 谭营 等译. 计算智能理论 [M]. 北京:清华大学出版社, 2010:130-131.
- [5] 李柏年 吴礼斌 著 MATLAB 数据分析方法 [M] 机械工业出版社, 2011:10-12.

【附录】

遗传算法代码（三次样条插值和多项式拟合、分段拟合要不停地改参数，附录中没有列出）

MyGA.m（程序入口）

```
function [ points,MinCost ] = MyGA( N,n,T,p,q )
%N为种群大小，T为迭代次数，n为每个个体基因个数，p为交叉概率，q为变异概率
Tic %开始计时
t=1;
CostArr=zeros(1,N);
data=xlsread('data.xlsx','A1:AY938');
group=Init(N); %产生初始种群
for i=1:1:N %计算个体成本
    CostArr(i)=CountCost(group(i,:),data);
end
MinCost=CostArr(N);
%模拟进化过程
while t<T
    %变异操作，经过子代合法性检查
    [group,CostArr]=Mutate(group,CostArr,N,q,data,n);

    %优势个体交配之前需要对成本进行降序排序，优势个体在下方
    [CostArr,Index]=sort(CostArr,'descend');
    group=group(Index,:);

    %优势个体交配，经过子代合法性检查
    [group,CostArr]=cross(group,CostArr,N,p,data,n);

    %淘汰劣势个体也要排序
    [CostArr,Index]=sort(CostArr,'descend');
    group=group(Index,:);

    %选择压
    [group,CostArr]=Select(group,CostArr,data,n);
    if MinCost>CostArr(N)
        MinCost=CostArr(N);
        points=group(N,:);
    end
    disp(MinCost);
    disp(group(N,:));

    disp('new generation!');
    disp(t);
```

```

        t=t+1;
    end
    toc %记录时间
end

```

Init.m

```

function [ group ] = Init(N,n) %产生初始N个个体，每个个体n个不同基因
group=ones(N,n);
for i=1:N
    group(i,:)=group(i,:)+randperm(49,n);
end

```

Mutate.m

```

function[ group,CostArr ]=Mutate( group,CostArr,N,q,data,n) %变异
n=6;
for i=1:1:N
    if rand<q
        while 1
            if CostArr(i)<(n*20+10) %经验表明精度造成成本为个位数，所以在
                % n*20+10时减小变异幅度
                delt=randi([-2,2]);
            else
                delt=randi([-5,5]);
            end
            Num=randi(n); %随机单点突变
            child=group(i,:);
            child(1,Num)=mod(group(i,Num)+delt,49)+2;

            %判断子代合法性，如果不同的基因座上有相同基因，则一直变异直到所有基因不同
            if length(unique(child))==n
                Cost=CountCost(child,data,n);
                if CostArr(i)>Cost; %子代更优，则取代亲代
                    group(i,:)=child;
                    CostArr(i)=Cost;
                end
                break;
            end
        end
    end
end
end

```

cross.m

```

function [group,CostArr] = cross( group,CostArr,N,p,data,n) %优势个体交配
n=6;
for i=N:-2:(N/2+2)
    if rand<p
        %随机单点交叉
        point=randi(n);
        child1=[group(i,1:point),group(i-1,point+1:n)];
        child2=[group(i-1,1:point),group(i,point+1:n)];
        %判断子代合法性，若出现不同基因座上有相同基因，取消交叉
        if (length(unique(child1))==n )&&(length(unique(child2))==n )
            Cost1=CountCost(child1,data,n);
            Cost2=CountCost(child2,data,n);

            if CostArr(i)>Cost1
                group(i,:)=child1;
                CostArr(i)=Cost1;
            end
            if CostArr(i-1)<Cost2
                group(i-1,:)=child2;
                CostArr(i-1)=Cost2;
            end
        end
    end
end
%适应度低的与适应度高的进行交叉，下标从N/2到1
for i=N/2:-1:1
    sort(group(i,:));
    if rand<p
        point=randi(n);
        child1=[group(i,1:point),group(i+N/2,point+1:n)];

        if (length(unique(child1))==n )
            Cost1=CountCost(child1,data,n);
            if CostArr(i)>Cost1
                group(i,:)=child1;
                CostArr(i)=Cost1;
            end
        end
    end
end
end
end

```

Select.m

```

function [ group, CostArr] = Select( group, CostArr, data, n)
%适应度低的相当于被移民取代，移民的是原始种群
group([1,2,3,4],:)=Init(4,n);
CostArr(1)= CountCost(group(1,:), data, n);
CostArr(2)= CountCost(group(2,:), data, n);
CostArr(3)= CountCost(group(3,:), data, n);
CostArr(4)= CountCost(group(4,:), data, n);
end

```

CountCost.m

```

function [ PointsCost ]=CountCost( points, data, my_answer_n) %成本计算函数
PointsCost=0;
sort(points);
[M,N]=size(data);
nsample=M/2; npoint=N;
X=zeros(nsample, npoint);
Y=zeros(nsample, npoint);
Ybar=zeros(nsample, npoint);
for i=1:nsample
    X(i,:)=data(2*i-1,:);
    Y(i,:)=data(2*i,:);
end
my_answer_gene=zeros(1, npoint);
my_answer_gene(points)=1;

% 定标计算
index_temp=logical(my_answer_gene);
X_optimal=X(:, index_temp);
Y_optimal=Y(:, index_temp);
for j=1:nsample
    % 3次样条插值
    Ybar(j,:)=interp1(X_optimal(j,:), Y_optimal(j,:), X(1,:), 'spline');

    %3次拟合
    % a=polyfit(X_optimal(j,:), Y_optimal(j,:), 3);
    % Ybar(j,:)=X(1,:).^3*a(1)+X(1,:).^2*a(2)+X(1,:)*a(3)+a(4);

    %4次拟合
    %a=polyfit(X_optimal(j,:), Y_optimal(j,:), 4);
    %Ybar(j,:)=X(1,:).^4*a(1)+X(1,:).^3*a(2)+X(1,:).^2*a(3)+X(1,:)*a(4)+a(5);

```

```

%5次拟合
%a=polyfit(X_optimal(j,:),Y_optimal(j,:),5);
%Ybar(j,:)=X(1,:).^5*a(1)+X(1,:).^4*a(2)+X(1,:).^3*a(3)+X(1,:).^2
*a(4)+X(1,:)*a(5)+a(6);

end

% 成本计算
Q=12;
errabs=abs(Y-Ybar);
le0_5=(errabs<=0.5);
le1_0=(errabs<=1);
le2_0=(errabs<=2);
le3_0=(errabs<=3);
le5_0=(errabs<=5);
g5_0=(errabs>5);
sij=0.5*(le1_0-le0_5)+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;
PointsCost=sum(si)/nsample;
end

```

test_ur_answer.m

```

PointsCost=0;
sort(points);
[M,N]=size(data);
nsample=M/2; npoint=N;
X=zeros(nsample,npoint);
Y=zeros(nsample,npoint);
Ybar=zeros(nsample,npoint);
for i=1:nsample
    X(i,:)=data(2*i-1,:);
    Y(i,:)=data(2*i,:);
end
my_answer_gene=zeros(1,npoint);
my_answer_gene(points)=1;

% 定标计算
index_temp=logical(my_answer_gene);
X_optimal=X(:,index_temp);
Y_optimal=Y(:,index_temp);

```

```

for j=1:nsample
    % 3次样条插值
    Ybar(j,:)=interp1(X_optimal(j,:),Y_optimal(j,:),X(1,:), 'spline');

    %3次拟合
    % a=polyfit(X_optimal(j,:), Y_optimal(j,:), 3);
    % Ybar(j,:)=X(1,:).^3*a(1)+X(1,:).^2*a(2)+X(1,:)*a(3)+a(4);

    %4次拟合
    %a=polyfit(X_optimal(j,:),Y_optimal(j,:),4);
    %Ybar(j,:)=X(1,:).^4*a(1)+X(1,:).^3*a(2)+X(1,:).^2*a(3)+X(1,:)*a(
    4)+a(5);

    %5次拟合
    %a=polyfit(X_optimal(j,:),Y_optimal(j,:),5);
    %Ybar(j,:)=X(1,:).^5*a(1)+X(1,:).^4*a(2)+X(1,:).^3*a(3)+X(1,:).^2
    *a(4)+X(1,:)*a(5)+a(6);

end

% 成本计算
errabs=Y-Ybar;
delt=zeros(469,1);
for i=1:469
    del(i,:)=lillietest(errabs(i,:));
end
disp(sum(del));

```