统计推断在数模转换系统中的应用

第 30 组　殷柯 5121109016，　王冠群 5130309301

摘要：本文在大量样本数据的基础上论述了为某种产品校准工序提供了优化方案的方法，探讨了统计推断在数模模数转化中的应用，以降低校准陈本并且提高精度为原则，综合运用曲线拟合，数理统计方法、启发式搜索等科学方法研究了产品输入输出的关系曲线。实践表明，此方法可以提供较优的校准方案，以满足工程实践的需要。

关键词：遗传算法，多项式拟合，三次样条插值，假设检验，MATLAB

Application of Statistical Inference in ADC system

ABSTRACT: This article is about providing an optimization method for a calibration scheme of some product on the basis of massive sample data with discussion of digital-analog modulus conversion applications. On the principle of reducing cost of calibration and improving the accuracy, we study the input-output relationship by using some scientific research approaches such as curve fitting, mathematical statistics, heuristic algorithm. The experimental results show that this method can provide a better calibration scheme which meets the demand of engineering practice.

Key words: GA, Polynomial Fitting, Cubic Spline Interpolation, Hypothesis Test, MATLAB

1 引言

在工程实践中，对某些物理量（如温度、压力、光强等）的测量需要用到特殊的测量工具，这些测量工具主要由传感器组成。由于批量生产，这些测量工具的输入输出特性会有差异。为了使测量结果尽可能准确，需要对测量工具进行定标。事实上，该输入输出特性往往呈现明显的非线性，因此为了得到较精确的结果，一种办法是尽可能多地选取测试点定标。但是，为了降低定标的成本，不可能对每一个测试点进行定标，因此需要合理地选择某些点进行定标，由这些点拟合出来的特性曲线近似替代其真实的特性曲线。合理的方案既要考虑尽可能减小拟合与真实值之间的误差，又要减少由定标测试点的个数带来的定标成本。本文就上述问题，探讨了如何选取测定点的数量和位置，以及合理的拟合方法得到输入输出的特性，使得在一定精度要求下，尽可能降低定标耗费的成本。

2 案例简述

本文以一个具体的案例，来探讨测量工具定标方案的实现。假设 X 代表输入，Y 代表输出，则输入输出特性主要有以下特征：Y 随 X 单调递增，X 在[5.0,10.0]区间内每隔 0.1 取值，Y 在[0,100]区间内取值，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。假设前期已经通过试验性小批量生产了一批测量工具样品，并通过实验测定了每个样品的特性数值。数据被绘制成表格，作为标准样本数据库。

对于该案例的每一个定标方案，都有一个成本函数来评价该方案的好坏。成本函数有测定成本和误差成本两个因素组成。单点误差成本记为$s_{i,j}$，表示第 i 个样本中的第 j 个数据点的拟合值与真实值之间的误差，若单点测定成本记为 q，则单个样本的定标成本为$S_i = \sum_{j=1}^{51} s_{i,j} + q \cdot n_i$，$n_i$ 表示对该样本个体定标过程中的单点测定次数，继而校准方案总体平均成本为$C = \frac{1}{M}\sum_{i=1}^{M} S_i$，M 是样本数。C 的大小就决定了该方案的好坏。

3 方案设计

本案例需要选择测定点的组合方案，相当于解一个组合优化问题，而由于组合方案数目庞大，是典型的 NP-hard 问题，因为计算时间过长，目前的计算机尚难以通过暴力穷举的方法，寻找出最优的测定点组合方案。因此，本文考虑采用遗传算法进行优化搜索。对测定点的拟合方法本文选用三次样条差值进行拟合。

3.1 遗传算法

3.1.1 遗传算法概述[1]

遗传算法（Genetic Algorithms，简称 GA）是一种基于自然选择原理和自然遗传机制的搜索算法，它是模拟自然界中的生命进化机制，在人工系统中实现特定目标的优化。遗传算法的实质是通过群体搜索技术，根据适者生存的原则逐代进化，最终得到最优解或准最优解。它必须做以下操作：初始群体的产生、求每一个体的适应度、根据适者生存的原则选择优良个体、被选出的优良个体两两配对，通过随机交叉其染色体的基因并随机变异某些染色体的基因后生成下一代群体，按此方法使群体逐代进化，直到满足进化终止条件。其实现方法如下：

（1） 根据具体问题确定可行解域，确定一种编码方法，能用数值串或字符串表示可行解域的每一解。

（2） 对每一解应有一个度量好坏的依据，它用一函数表示，叫做适应度函数，适应度函数应为非负函数。

（3） 确定进化参数群体规模 、交叉概率 、变异概率 、进化终止条件。

3.1.2 遗传算法在本案例中的实现

（1）编码： 采用 01 编码，对于输入数据 X 的某一段取值区间中，每一个 X 的取值对应 0 或 1,1 代表该点取为测定点，0 代表不取为测定点。由常识知这一段区间中的数据第一个点和最后一个点须取。例如 101001，代表第 1 个，第 3 个，第 6 个位置是取点的，其他位置不取。编码为 1 的点取为测定点进行定标，并且由这些点拟合输入输出特性曲线。

（2）选取初始群体： 对于输入数据 X 的某一段取值区间，随机产生 01 串（保证首位和末位为 1）作为一个解的个体，然后按照同样的方法产生其他个体，初始群体的大小暂定为 100。这些个体构成一个 01 矩阵，矩阵的每一行即为对应的第几个个体解。

（3）适应度函数： 对每一个个体解，可以在输入数据 X 的某一段取值区间拟合出一条曲线，然后和真实值比较，可以得到该区间上样本总体的平均成本，而适应度函数即为平均成本的倒数。适应度函数的大小反应了个体解的生存几率，适应度函数大的个体解生存几率大，反之相反。

（4）种群选择： 一个群体 100 个个体，每个个体有一个适应度函数值，代表生存可能，

把适应度加和。100 个适应度函数值把适应度总和划分成了 100 个子区间，然后利用计算机产生 0 到适应度和之间的随机数，该随机数落在哪个子区间，就表示哪个对应的个体解被选为种群中的个体。这样组成一个 100 行的 01 矩阵。这一步决定了遗传的方向是向优的方向进行的。

（5）交叉： 交叉概率乘上种群个体数即为发生交叉的组数，种群矩阵中的个体解随机配对，组成一组交叉组，在每个交叉组中，随机选取个体解中的某一个 01 位，该位后面的 01 编码则两两交换，得到新的矩阵。

（6）变异： 在交叉得到的矩阵中，每一位 01 字符以变异率率取补（0 变 1,1 变 0），即产生[0,1]上的随机数，若小于变异率则该位取补，但是每一行的解个体的首位和末位是不变的，恒为 1。

（7）循环： 变异得到的新矩阵就是下一次循环开始的初始群体，返回上述第（3）步，再次进入循环。记录每次循环产生的最优个体解。循环的终止条件为循环 500 代（若不再产生更优的解可提前终止）。代码见附录。

上述算法思想用算法流程图表示如下：

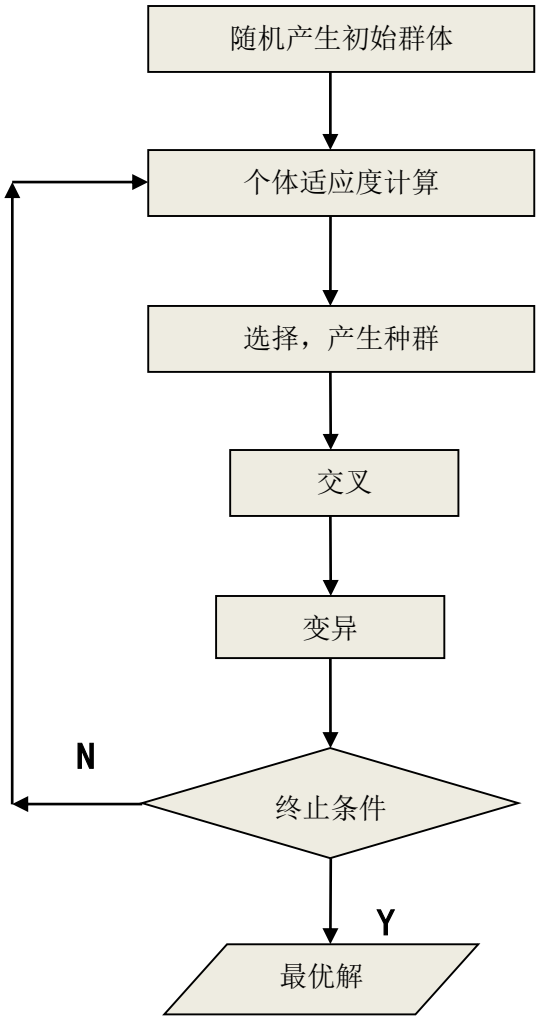


图 3-1-1 遗传算法流程图

3.1.3 遗传算法的改进

（1）遗传算法的方向是由适应度函数控制的，一代代的解总的效果是使成本降低，但由于随机性，下一代的解可能不如上一代优，因此循环终止时的种群的最优解未必是整个进化过程中的最优解，所以每一次循环都需要记录该代的最优解，并输出显示，以便监测循环。

（2）遗传算法中的选择，交叉，变异都有可能把种群矩阵中的最优的个体破坏，导致下一代的最优解不如上一代，为了让进化总是向优的方向进行，每一次操作前，可以把上一次操作的最优解保护起来，等操作完成后，将上一次操作的最优解与这次的比较，取其中更优的一个解再放入种群矩阵。

（3）遗传算法的交叉概率通常为0.5，变异概率通常为0.001，为了提高收敛速度，可以适当调整这两个值，以增加或减少交叉和变异数量。

（4）当运行过几次代码后，对测定点的个数和成本之间的关系有个大致了解，可以在随机产生初始种群的过程中加入一些干扰，例如限制取点的个数，使初始种群就具有较优的个体，提高收敛速度。

3.2 三次样条插值

三次样条插值法是数学上的一种拟合方法。从数学角度分析，在每个区间$[x_k, x_{k+1}]$可构造一个三次函数，使得分段曲线$y = S(x)$和它的一阶导数和二阶导数在更大的区间$[x_0, x_N]$内连续。$S'(x)$的连续性意味着曲线$y = S(x)$没有急弯。$S''(x)$的连续性意味着每点的曲率半径有定义。三次样条插值的具体数学定义[2]如下：

设$\{(x_k, y_k)\}_{k=0}^{N}$有$N+1$个点，其中$a = x_0 < x_1 < \cdots < x_N = b$。如果存在N个三次多项式$S_k(x)$，系数为$s_{k,0}, s_{k,1}, s_{k,2}, s_{k,3}$，满足如下性质：

(1) $S(x) = S_k(x) = s_{k,0} + s_{k,1}(x - x_k) + s_{k,2}(x - x_k)^2 + s_{k,3}(x - x_k)^3$，

$x \in [x_k, x_{k+1}]$且 $k = 0,1,\cdots, N-1$ (3-2-1)

(2) $S(x_k) = y_k$， $k = 0,1,\cdots, N$ (3-2-2)

(3) $S_k(x_{k+1}) = S_{k+1}(x_{k+1})$， $k = 0,1,\cdots, N-2$ (3-2-3)

(4) $S_k'(x_{k+1}) = S_{k+1}'(x_{k+1})$， $k = 0,1,\cdots, N-2$ (3-2-4)

(5) $S_k''(x_{k+1}) = S_{k+1}''(x_{k+1})$， $k = 0,1,\cdots, N-2$ (3-2-5)

则称函数$S(x)$为三次样条函数。

MATLAB 中自带了三次样条插值函数 spline 用于拟合。

3.3 两种可行思路

方案一，测量工具的输入 X 的取值区间为整段数据点，即[5.0,10.0]，对该区间运用遗传算法和三次样条插值，选取测定点的组合方案，使平均成本最小。

方案二，由于测量工具的特性曲线呈现明显的三段，因此考虑分段拟合并求成本，因此首先要确定中段起止点位置。中段的起止位置确定以后，就分前中后三段分别以三次样条插值法拟合，每一段用遗传算法选取优化的点的位置和个数，计算成本，然后三段成本加和，再扣除重复计算的中段起止点的测定成本。

4 模型求解

4.1 方案一求解

方案一，以整段数据点为研究对象，运用遗传算法和三次样条插值，选取测定点的组合方案，运行代码，求得最优解，较好的三次运行结果如下：

表4-1-1 方案一运行结果

测定点的组合方案（点的序号）	平均成本
1， 9， 15， 26， 31， 42， 51	100.37
1， 12，21， 28， 37， 45， 51	97.99
1， 9， 20， 26， 34， 44， 51	95.21

4.2 方案二求解

4.2.1 确定中段的起止点位置

首先，需要取得每组中段的近似起止点位置，取几组数据观察，进行三次多项式拟合时MATLAB 作图如下（*代表原始数据，实线代表拟合曲线）：

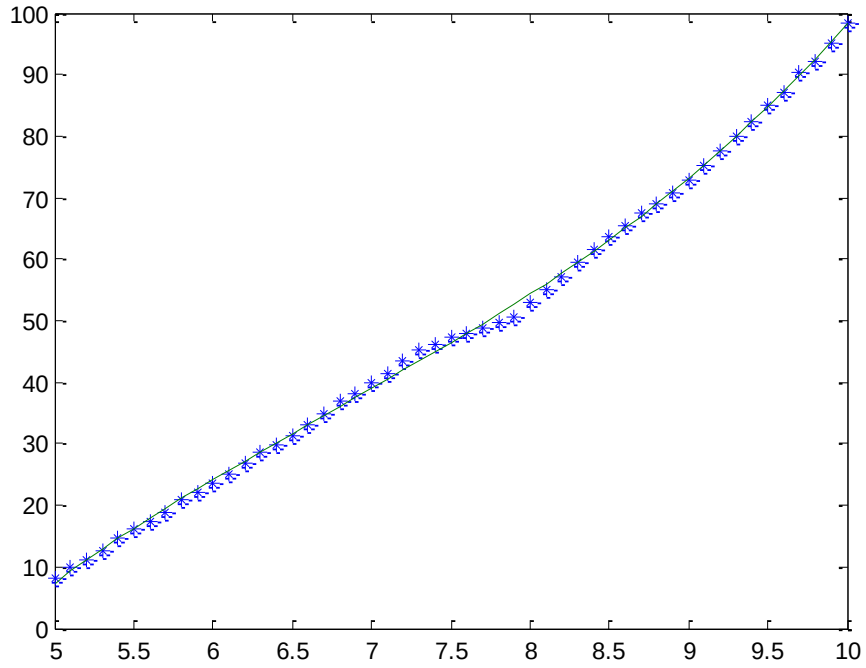


图 4-2-1 某组数据三次多项式拟合数据和原始数据点的比较图

由图可知，虽然三次多项式的拟合不很准确，但给中段起止点位置的选择带来方便。起始点可以近似看成原始数据曲线比拟合曲线高出最多的一个点，终止点可以看成原始数据曲线比拟合曲线低得最多的一个点。对应于向量表示，**x**:5.0-10.0 的横坐标向量，**y**：原始数据的纵坐标向量，**y1** 拟合产生的对应纵坐标向量，则向量 **y - y1** 元素的最大值对应的点就是起始点，最小值对应的点就是终止点。把这两点的横坐标取出来就是这一组的起止点位置，对 469 组做同样操作，然后得到起始点横坐标向量 **x1**，终止点横坐标向量 **x2**。具体过程见附录代码。

其次，对得到的 469 组起止点位置，考察它们的概率分布，可以有两种办法(以起始点为例)：

（1）469 个不同的起始点构成一个样本，由样本推断其真实总体是否满足某种概率分布，此处假设满足正态分布，然后如果经过非参数假设检验验证满足正态分布，则由总体的均值

μ，近似代表起始点位置。然而x~N（μ，σ^2）中，μ 和σ^2均未知，可以用 469 个起始点的样本对总体进行参数估计，得到μ和σ的最大似然估计值，并用μ作为起始位置。其中，非参数检验[3]采用非参数χ^2检验，步骤如下：

步骤 1：将样本空间Ω划分成 k 个互不相容的事件A1，A2，……Ak；

步骤 2：在假设H_0为真时，计算该分布的理论概率p_i=P(Ai)，$i = 1,2,\cdots$，k；

步骤 3：由实验数据确定事件 Ai 发生的频率 $\frac{fi}{n}$；

步骤 4：采用统计量$\chi^2 = \sum_{i=1}^{k} \frac{(fi-npi)^2}{npi}$,则近似地有$\chi^2 = \sum_{i=1}^{k} \frac{(fi-npi)^2}{npi} \sim \chi^2(k-r-1)$,r 为分布函数中被估计参数的个数，在$H_0$成立的条件下，水平为α的拒绝域为 $W = [\chi_\alpha^2(k-r-1), +\infty)$。

由此对向量 x1 做非参数假设检验，列表如下：

表 4-2-1 x1 的非参数检验表

A_i	f_i	p_i	np_i	$f_i - np_i$	$(f_i - np_i)^2/ np_i$
[5, 6)	5	0.0368	17.2592	-12.2592	8.7077
[6, 6.5)	0	0.1106	51.8714	-51.8714	51.8714
[6.5, 7)	152	0.2309	108.2921	43.7079	17.6410
[7, 7.5)	210	0.2868	134.5092	75.4908	42.3678
[7.5, 8)	81	0.2122	99.5218	-18.5218	3.4470
[8, 8.5)	0	0.0934	43.8046	-43.8046	43.8046
[8.5 10)	21	0.0285	13.3665	7.6335	4.3594

易知 k=7,水平为α=0.05 的拒绝域为$[9.488, +\infty)$,上表最后一列求和显然在拒绝域内，因此起始点数据不满足正态分布，实际尝试中还选取[6.5,8)间较密集的数据做了一遍，仍不满足正态分布。具体代码见附录

(2) 当起始点横坐标数据不满足某一分布时，直接用样本的均值代替总体均值，作为起始点位置，即向量 x1 和 x2 的均值分别为 7.2 和 7.9，作为起止点位置（对应编号为 23 和 30）。

4.2.2 分段求解

由上述讨论，将整段数据点分为前中后三段，编号分别为 1-23,23-30,30-51，对应的 X 区间为[5.0,7.2],[7.2,7.9],[7.9,10.0]。三段分别以三次样条插值法拟合，每一段用遗传算法选取优化的点的位置和个数，计算成本，然后三段成本加和，再扣除重复计算的中段起止点的测定成本 12*2=24。三段的代码运行结果如下：

表4-2-2 方案二运行结果

	测定点的组合方案（点的序号）	平均成本
前段	1， 13， 23	40.2719
中段	23， 30	26.8198
后段	30， 41， 51	43.1407
总和	1， 13， 23， 30， 41， 51	86.2324

5 拓展——拟合方法的选择

本文主要采用三次样条插值作为拟合方法，应用于整段求解和分段求解的情况，实际上拟合方法还可以是多项式拟合，指数拟合，同时分段的每一段也可有不同的拟合方法，因此就有了很多的拟合方法的组合。本案例以多项式拟合为例，再次利用遗传算法选取测定点组合方案，用于对上述方案的补充和对比。另外多项式拟合，需分情况，依取点个数，选择多项式的次数。例如，3 个点选取二次多项式拟合，4 个点选三次多项式拟合，大于等于 6 个点，用五次多项式拟合。代码上，须在单组成本函数 cost.m 中做出相应修改。由于上述分段情况下，每段的选点个数是 2-3 个点，因此此时在 MATLAB 中三次样条插值 spline 的拟合为，二次和一次多项式，与直接多项式 polyfit[4]拟合本质上是一样的，故只讨论多项式拟合应用在整段的情况。当采用遗传算法和多项式拟合时的运行结果如下（取其中较好的三次运行结果）：

表5-1 多项式拟合运行结果

测定点的组合方案（点的序号）	平均成本
1， 6， 16， 24， 32， 42， 51	110.9861
1， 6， 19， 25， 33， 44， 51	112.581
1， 4， 14， 23， 31， 45， 51	112.3092

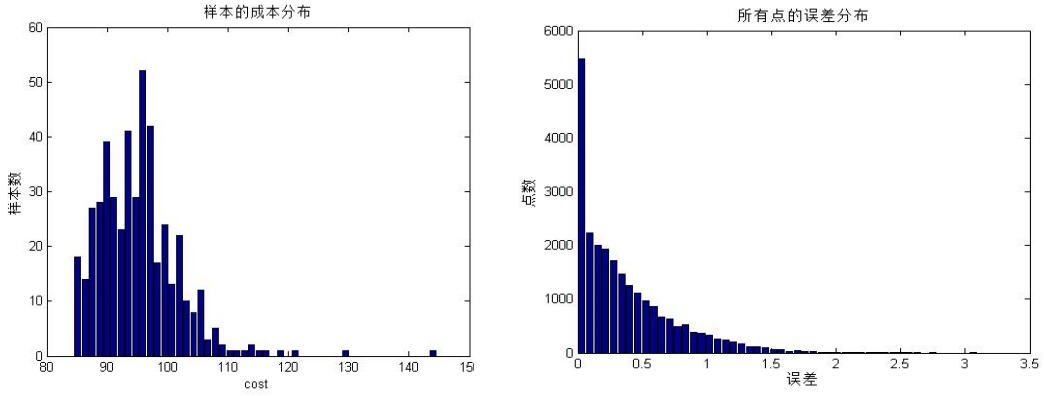
6 模型评价

本文主要有三个模型，三次样条插值法的分段拟合，三次样条插值法的整段拟合，还有多项式的整段拟合。评价一个模型的好坏，可以从运行结果，取点个数带来的标定成本大小，误差大小，运行效率等角度考察。

从运行的结果来看，三次样条函数的分段拟合的成本最低，为 86.2324，三次样条插值法的整段拟合里最好的一次结果成本其次，为 95.21，多项式的整段拟合效果不佳，成本是 110.9861。

从取点个数来看，三次样条插值的分段选了 6 个点，其他两种方案选了 7 个点，分段的方案标定成本更低。

从误差大小来看，三次样条插值法的整段拟合（最好的一次运行结果），三次样条插值法的分段拟合，还有多项式的整段拟合（最好的一次运行结果）的各类误差统计图如下：



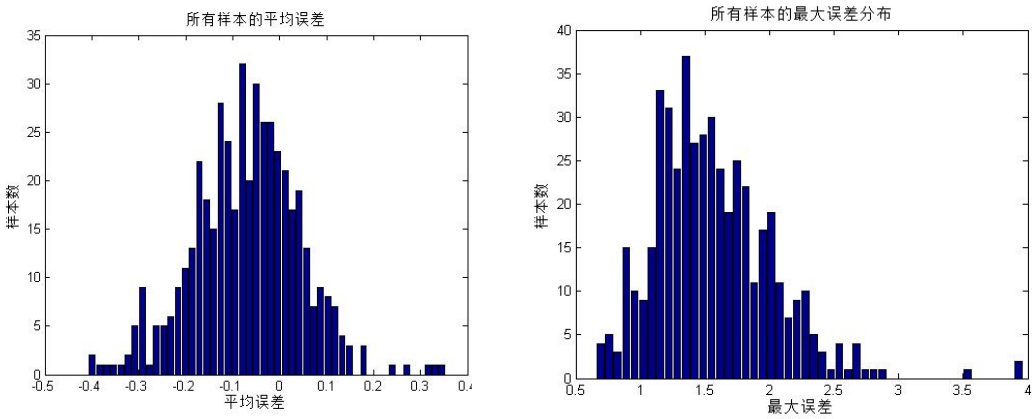


图 6-1 三次样条插值整段拟合各类误差统计图

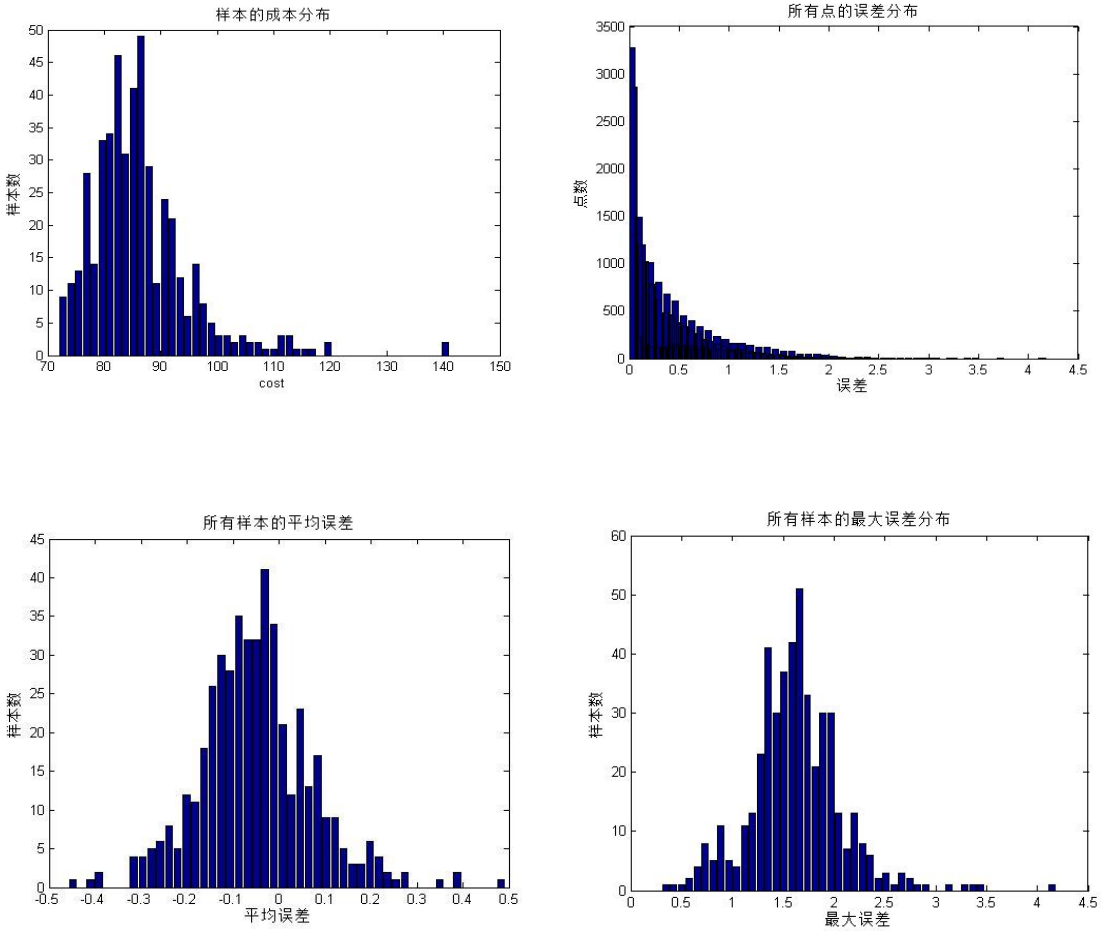


图 6-2 三次样条插值分段拟合各类误差统计图

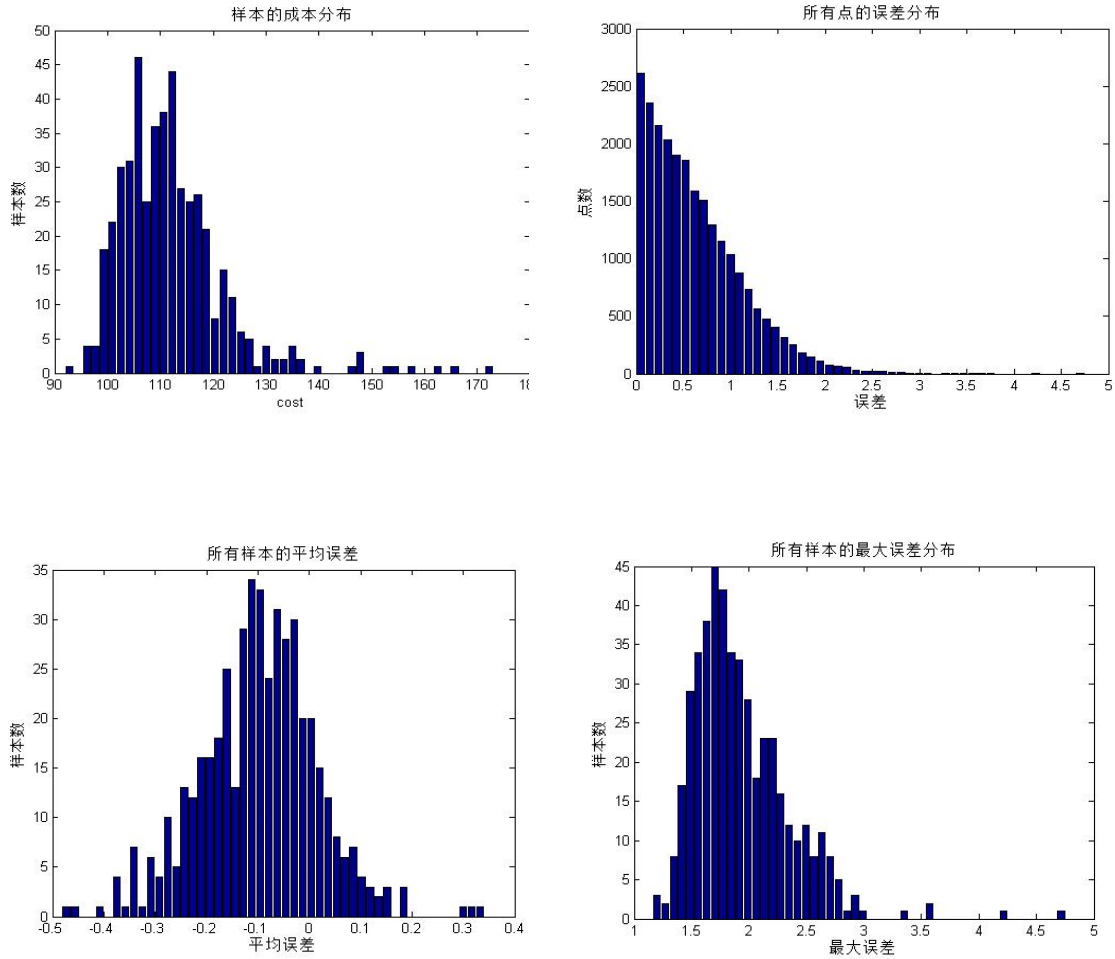


图 6-3 多项式整段拟合各类误差统计图

由上图比较可知，三次样条插值分段拟合较其他两种方案样本成本分布，样本成本分布重心更小，所有点的误差分布和所有样本的平均误差更靠近 0，最大误差分布中在大误差处有更少的样本，因此，此分段方案更精确。这是由于三次样条过所有测定点，而多项式拟合不一定过测定点，偏差会大一些。另外，中段的起止点位置是最容易产生较大误差的点，分段选取这两个点为必测点，有效地减小了它们带来的误差。

最后，从实际运行效率来看，分段的时候，由于编码长度更短，代码运行更快，并且收敛更快，用较少的代数就可以得到分段上的最优解，而整段三次样条则运行慢一些，整段的多项式则运行代数很多，不易得到成本更低的解。

综上，提供校准方案时，三次样条插值法的分段拟合是最有效的方案。

7 结论

本文主要采用三次样条插值和遗传算法对案例的取点方案进行优化搜索，并辅以多项式拟合作为拓展补充，在所得的全部运行结果中，分段三次样条插值的平均成本最低，为 86.2324。具体的测定点组合方案如下：测定点编号为 1，13，23，30，41，51，对应 X 为 5.0，6.2，7.2，7.9，9.0，10.0。拟合方法为，前段[5.0,7.2],中段[7.2,7.9],后段[7.9,10.0],三段分别用三次样条插值进行拟合。

8 参考文献

[1] 算法大全[M].http://wenku.baidu.com/link?url=5c-XOHZRWDEiTMHCzV5X_4mkNrP7rGofC-M0YiS3bvVSr0OnSxogGjrzJ5pFi0zpWjUA_TtXGmMJOQc1-nwreHY_5QMZK0jZSAPux9Txtpe

[2] John H.Mathews ,Kurtis D.Fink 著，陈渝，周璐，钱方,等译.数值方法（MATLAB 版）[M].北京：电子工业出版社,2002:209-210.

[3] 武爱文,冯卫国，卫淑芝，熊德文，等. 概率论与数理统计[M]. 上海:上海交通大学出版社,2011:250-251.

[4] 王沫然. Matlab6.0 与科学计算[M]. 北京:电子工业出版社,2001:70-71..

[5] 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.

9 附录

附录 1：三次样条插值单组成本函数（整段和分段做相应修改）（MATLAB）

```
function f= cost(x,y,group )
j=0;everycost=0;start=1;final=51;count=final-start+1;
choosex=zeros(1,sum(group));
choosey=zeros(1,sum(group));
for i=1:count
    if(group(i)==1)
        j=j+1;
        choosex(j)=x(i);
        choosey(j)=y(i);
    end
end

y1=spline(choosex,choosey,(5.0:0.1:10));%三次样条插值

d=zeros(1,count);
for i=1:count
    d(i)=abs((y(i)-y1(i)));
end
for i=start:final
    if(d(i)<=0.5)
        everycost=everycost+0;
    end
    if(d(i)>0.5&&d(i)<=1)
        everycost=everycost+0.5;
    end
    if(d(i)>1&&d(i)<=2)
        everycost=everycost+1.5;
    end
```

```
    if(d(i)>2 && d(i)<=3)
        everycost=everycost+6;
    end
    if(d(i)>3 && d(i)<=5)
        everycost=everycost+12;
    end
    if(d(i)>5)
        everycost=everycost+25;
    end
end
f=everycost+12*sum(group);
end
```

附录 2:三次样条插值样本总体平均成本函数(整段和分段做相应修改)(MATLAB)

```
function y = avgcost(A,group)
start=1;final=51;
sum=0;
for i=1:469
    x=A(2*i-1,start:final);
    y=A(2*i,start:final);
    sum=sum+cost(x,y,group);
end
y=sum/469;
end
```

附录 3:遗传算法主程序(整段和分段做相应修改)(MATLAB)

```
function   f= geneticalgorithm()
warning off all
N=100;%种群数
generation=500;%循环代数
rateofcross=0.5;%交叉率
rateofchange=0.01;%变异率
start=1;%起始点
final=51;%终点
count=final-start+1;%编码长度
A=xlsread('20141010dataform.csv');
allgroup=zeros(N,count);%种群矩阵
allgroupfitness=zeros(1,N);%种群适应度矩阵
tmpvalue=852;
tmpvector=zeros(1,count);
num=2;%取点个数

%初始群体产生
for j=1:N
```

```
    group=zeros(1,count);
    group(1)=1;group(count)=1;
    out=randperm(50);
    data=out(1:8);
    for i=1:8
        group(data(i))=1;

    end
    allgroup(j,:)=group;
end

costvector=zeros(1,N);
for i=1:N
    costvector(i)=avgcost(A,allgroup(i,:));
end

[value,id]=min(costvector);
tmpvalue=value;
tmpvector=allgroup(id,:);


for t=1:generation%循环代数

for k=1:N%计算适应度
    allgroupfitness(k)=1/avgcost(A,allgroup(k,:));
end

sumoffitness=zeros(1,N);%适应值累加
sumoffitness(1)=allgroupfitness(1);
for i=2:N
    sumoffitness(i)=sumoffitness(i-1)+allgroupfitness(i);
end


groupafterchoose=zeros(N,count);%选择
for i=1:N
    tmp=unifrnd(0,sumoffitness(N));%随机产生0~适应度和的一个double数

    if(tmp<=sumoffitness(1))
            groupafterchoose(i,:)=allgroup(1,:);
    end

    for j=1:(N-1)
        if(tmp>sumoffitness(j)&&tmp<=sumoffitness(j+1))
```

```
            groupafterchoose(i,:)=allgroup(j+1,:);%产生选择后的种群
        end
    end
end

groupafterchoose(1,:)=tmpvector;


for i=1:rateofcross*N%交叉5组
    cut=start+randi(count-1)-1;%交叉位1~50
    line1=randi(N);
    line2=randi(N);
    tmp=groupafterchoose(line1,cut+1:final);%
    groupafterchoose(line1,cut+1:final)=groupafterchoose(line2,cut+1:final);
    groupafterchoose(line2,cut+1:final)=tmp;
end

%选择出交叉后的最优个体，并与先前的最优个体比较
costvector=zeros(1,N);
for i=1:N
    costvector(i)=avgcost(A,groupafterchoose(i,:));
end
[value,id]=min(costvector);

if value>tmpvalue
    groupafterchoose(1,:)=tmpvector;
end

if value<=tmpvalue
    tmpvalue=value;
    tmpvector=groupafterchoose(id,:);
end

groupaftercross=groupafterchoose;%交叉完毕


for i=1:N%变异
        for j=(start+1):(final-1)
            tmp=unifrnd(0,1);%0~1随机数
            if(tmp<=rateofchange&&groupaftercross(i,j)==1)
                groupaftercross(i,j)=0;
            end
            if(tmp<=rateofchange&&groupaftercross(i,j)==0)
                groupaftercross(i,j)=1;
```

```
                end
            end
end
%选择出变异后的最优个体，并与先前的最优个体比较
costvector=zeros(1,N);
for i=1:N
    costvector(i)=avgcost(A,groupaftercross(i,:));
end
[value,id]=min(costvector);

if value>tmpvalue
    groupaftercross(1,:)=tmpvector;
end

if value<=tmpvalue
    tmpvalue=value;
    tmpvector=groupaftercross(id,:);
end

allgroup=groupaftercross;%变异完毕

disp(['Generation',num2str(t),';']);
disp([num2str(tmpvalue),'          ',num2str(tmpvector)]);
end

%最后一代

f=tmpvector;
end
```

附录4：求中段起始点的位置（MATLAB）

```
function f = turndot(A)
x1=zeros(1,469);
x2=zeros(1,469);
m1=0;
m2=0;
for i=1:469
    x=A(2*i-1,:);
  y=A(2*i,:);
    p=polyfit(x,y,3);
    y1=p(1)*x.^3+p(2)*x.^2+p(3)*x+p(4);
    [m1,index1]=max(y-y1);
   x1(i)=x(index1);
     [m2,index2]=min(y-y1);
```

```
        x2(i)=x(index2);
end
f=[x1;x2];
end
```

附录 5：对中段起始点的假设检验用到的代码

```
A=xlsread('20141010dataform.csv');
x=turndot(A);
x1=x(1,:);
x2=x(2,:);
f1=zeros(1,10);
f2=zeros(1,10);
for i=1:10
    for j=1:469
        if (x1(j)>=(4.5+i*0.5) & x1(j)<5+i*0.5 )
            f1(i)=f1(i)+1;
        end
        if(x2(j)>=(4.5+i*0.5) & x2(j)<5+i*0.5 )
            f2(i)=f2(i)+1;
        end
    end
end

for k=1:469
    if x1(k)==10
        f1(10)=f1(10)+1;
    end
    if x2(k)==10
        f2(10)=f2(10)+1;
    end
end

f1;
f2;

[u1,sig1,u1ci,sig1ci]=normfit(x1,0.05);
[u2,sig2,u2ci,sig2ci]=normfit(x2,0.05);
p1=zeros(1,10);
p2=zeros(1,10);
for n=1:10
    p1(n)=normcdf(5+n*0.5,u1,sig1)-normcdf(4.5+n*0.5,u1,sig1);
    p2(n)=normcdf(5+n*0.5,u2,sig2)-normcdf(4.5+n*0.5,u2,sig2);
end
```

```
p1;
p2;
```

附录6：多项式拟合单组成本函数（MATLAB）

```
function f= cost(x,y,group )
j=0;everycost=0;start=1;final=51;count=final-start+1;
choosex=zeros(1,sum(group));
choosey=zeros(1,sum(group));
for i=1:count
    if(group(i)==1)
        j=j+1;
        choosex(j)=x(i);
        choosey(j)=y(i);
    end
end

if(sum(group)==2)%多项式¨
    a=polyfit(choosex, choosey, 1);
    y1=x*a(1)+a(2);
end
if(sum(group)==3)
    a=polyfit(choosex, choosey,2 );
    y1=x.^2*a(1)+x*a(2)+a(3);
end
if (sum(group)==4)
    a=polyfit(choosex, choosey, 3);
    y1=x.^3*a(1)+x.^2*a(2)+x*a(3)+a(4);
end
if (sum(group)==5)
    a=polyfit(choosex, choosey, 4);
    y1=x.^4*a(1)+x.^3*a(2)+x.^2*a(3)+x*a(4)+a(5);
end
if(sum(group)>=6)
    a=polyfit(choosex,choosey,5);
    y1=x.^5*a(1)+x.^4*a(2)+x.^3*a(3)+x.^2*a(4)+x*a(5)+a(6);
end


d=zeros(1,count);
for i=1:count
    d(i)=abs((y(i)-y1(i)));
end
for i=start:final
    if(d(i)<=0.5)
```

```
        everycost=everycost+0;
    end
    if(d(i)>0.5&&d(i)<=1)
        everycost=everycost+0.5;
    end
    if(d(i)>1&&d(i)<=2)
        everycost=everycost+1.5;
    end
    if(d(i)>2 && d(i)<=3)
        everycost=everycost+6;
    end
    if(d(i)>3 && d(i)<=5)
        everycost=everycost+12;
    end
    if(d(i)>5)
        everycost=everycost+25;
    end
end
f=everycost+12*sum(group);
end
```