统计推断在数模模数转换中的应用

第 057 组　潘进 5130309429 任嘉祥 5130309428

摘要: 本次课题是研究某监测模块中传感器部件监测的对象的物理量与传输部件的输出电压信号之间的关系。本文则是通过对原始数据的研究，通过三次多项式和三次样条插值以及三次插值对数据进行拟合，最后根据平均分的好坏得到相对较佳的拟合曲线。

关键词: 多项式拟合，三次插值，三次样条插值，模拟退火算法

1 引言

本次课题研究的是某监测模块中传感器部件监测的对象的物理量 y 与传输部件的输出电压信号 x 之间的关系。由于元件等各方面因素，每个元件的 y 与 x 的对应关系都不相同，如果逐个的去分析各个监测模块的性质，必定会耗费大量的人力物力，故通过剔除一些点，选取某些具有代表性的点，从而确定出监测模块相应参数之间的关系，便引出了该课题研究。

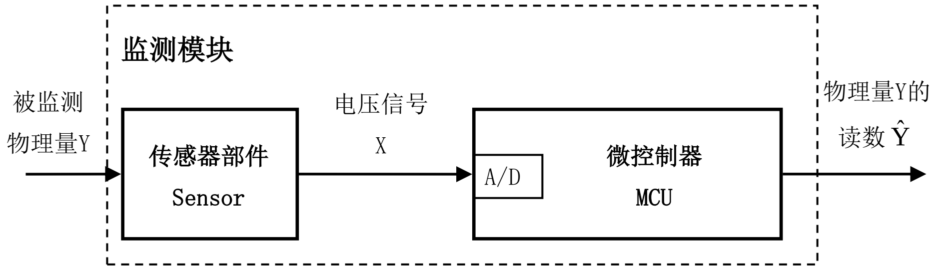


图 1 监测模块的组成框图

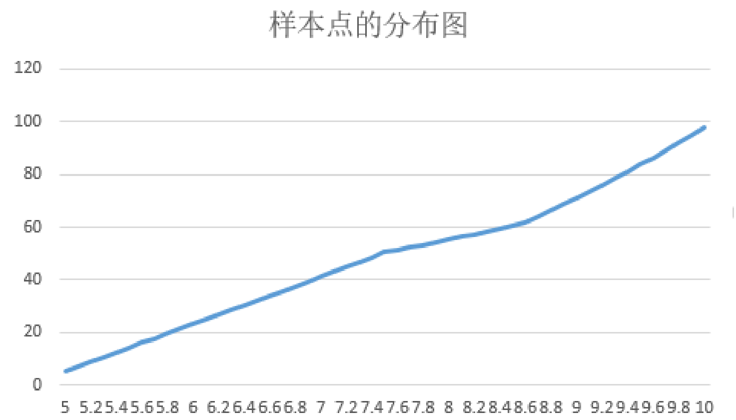
2 数据样本的分析

2.1 特征点的选取

因为需要测量的点有 51 个（按照每格 0.1 取点），故在工程实践中效率太低，所以我们倾向于选取典型的少数的点来推测 51 个点的 y 与 x 的关系。根据 Excel 表格做出的几个样本的点的分布，我们发现，在图像的首尾位置弯曲度比较大，而中间，即 6.0—9.0 位置的点差不多在一条直线上。故我们选择在第 0—10 个点内取 2 个点，在第 11—41 个点内取三个点，在第 42—51 个点内取 2 个点。具体的区间取法如下图：

表 1：三段曲线拟合区间划分

区间	电压范围	对应的采样点序号	特征点数目
1	[5.0V 5.9V]	1,2,…,10	2
2	[6.0V 9.0V]	11,12,…,41	3
3	[9.1V 10.0V]	42,43,…,51	2



2.2 数学模型的检验

通过上述选点得到监测模块的输入输出的各种关系特性。为了选出实际误差最小的模型，给出以下的检测函数。对任意给定的的对象，先测得一组观察值{ (x_{ij} ,y_{ij}) },其中 i=1,2,3,…,n ,n<51, 将 x_i 带入到拟合得到的函数当中得到(x_{ij} , y_{ij}),之后我们根据构建的精准度去评价函数。评价函数如下：

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

3 问题的拟合求解

3.1 多项式拟合

在该课题研究中，利用 matlab 的 polyfit 函数可以对数据进行高次多项式拟合，其中 polyfit 函数调用的形式为 polyfit(x,y,m)。表示的是用 m 次多项式来拟合数据 x 和 y。

拟合的表达式为$y = a_n x^n + a_{n-1} x^{n-1} + \ldots\ldots + a_0$ 。x 为自变量，表示输出电压信号，y 为 x 的函数，表示传感器部件监测的对象物理量。

我们随机产生一组点集{1，6，17，29，38，41，51}，并作出其三次拟合,四次拟合,五次拟合以及六次拟合的残差平方图，计算相应残差平方的平均值。

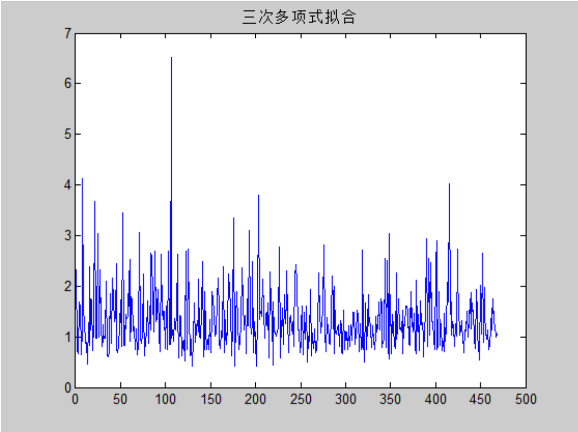
三次多项式拟合残差平方平均值：1.3344；

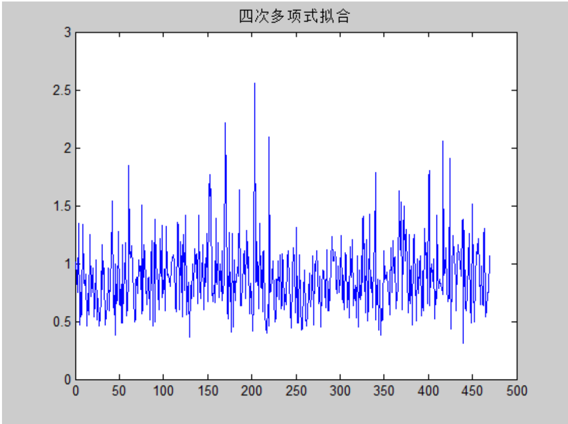
四次多项式拟合残差平方平均值：0.8855；

五次多项式拟合残差平方平均值：0.9835；

五次多项式拟合残差平方平均值：1.1311；

在随机选取四组点集，发现实验结果与上述结果类似，故在多项式拟合当中，四次多项式拟合更有代表性。





3.2 插值拟合

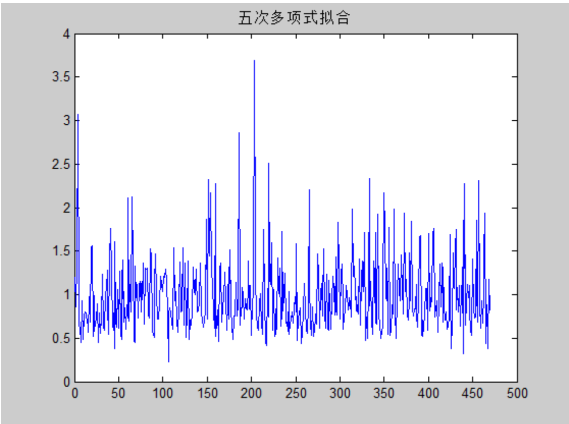
考虑构造一个过 $x_1,x_2,\ldots x_n$ 的次数不超过 n 的多项式 y=Ln(x)，使其满足 $Ln(x_k)=y_k$，k=1,2,3…n，然后用 Ln(a)作为准确值 L(a)的近似值。这种方法叫做插值。

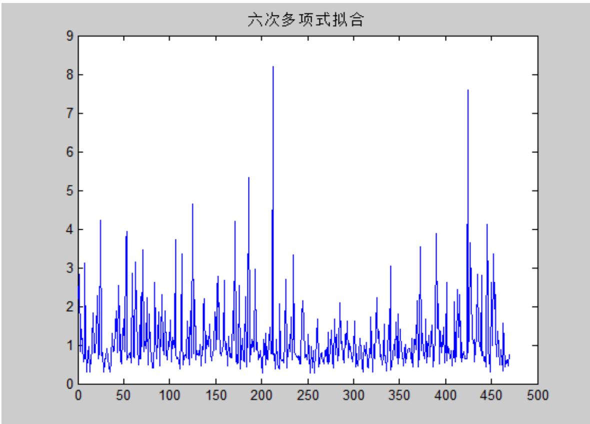
利用 matlab 中的一位数据的插值函数：interp1( )可以实现插值拟合。该函数的调用方式为 y_1=interp1(x,y,x_1,methods)。其中 x,y 分别表示一直数据点的横纵坐标，x_1 为插入的横坐标，methods 为可选参数，可分为以下四种：

(1) nearest——最近邻点插值；

(2) linear——线性插值（可缺省）；

(3) spline——三次样条插值；





(4) cubic——三次插值；

其中需要注意的是，该函数的调用前提的 x 必须为单调的且 x_1 不能超过 x 的值。

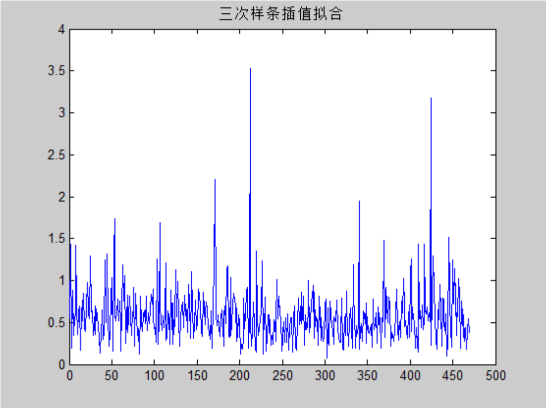
仍然由 3.1 中随机取得的点集{1，6，17，29，38，41，51}计算及作图，可得：

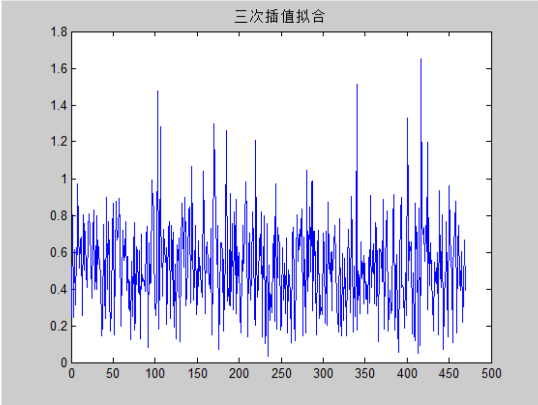
三次样条插值残差平方平均值为：0.5825；

三次插值残差平方平均值为：0.5276；

下图为利用三次样条插值和三次插值拟合得到的各组的残差平方和图像。

再随机选取四个点集，发现实验结果与上述结果类似，故初步判断三次样条插值和三次插值拟合比多项式拟合更具代表性。





4 模拟退火算法

本文 2-1 选用的特征点选取方法不准确，在确定了拟合的函数后，通过模拟退火方法，可以找到更具代表性的七个特征点。

模拟退火的原理：模拟退火算法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。退火算法模拟的是自然界中的降温过程，在一个温标 T 下，增加 Δt，分子的热运动平均能量增量 $\Delta E = -e^{k\Delta t/T}$，其中 E 为温度 T 时的内能，ΔE 为其改变量，k 为 Boltzmann 常数。

模拟退火算法核心思想：在温度较高的情况下，系统有较大的概率接受误差较大的分子（类比于温度较高时，各分子的动能波动范围大），此种模糊判断的好处是避免了系统过快地局部收敛。随着温度的降低，可接受变异误差范围逐渐减小，最终趋于稳定，即得出最优解

模拟退火算法的步骤：

(1) 随机产生一个初始解 S，并计算目标函数值 f(S,data)，此目标函数返回误差平方总和;

(2) 设置初始数值 TEPM=10.0，终止温度 TEMPEND=5.0 迭代次数 L= 6;

(3) do while TEMP > TEMPEND

1) for i=1~10，do while counter<6

2) 对当前最优解 S 按照某一邻域函数，产生一新的解 Snew。计算新的目标函数值 f(Snew,data) ，并计算目标函数值的增量 ΔE =f(S,data)-f(Snew,data)。

3) 如果 ΔE ＜0，则 S=Snew；

4) 如果 ΔE ＞0，则 p = exp(- ΔE /TEMP);

5) 如果 c = random[0,1] < p， S=Snew; 否则 S=S。

6) 如果 S=Sold，counter 加 1。

7) end do

8) end for

(4) i = i ＋ 1；

(5) end do

(6) 输出当前最优点，计算结束

我们通过使用 matlab 软件编写程序（程序见附录），分别应用不同的拟合算法的到相应拟合算法下最具代表性的七个点，并计算这七个点所需的成本。

其中，用三次插值进行拟合得到的最优点集为{3,13,21,27,34,43,50},其成本为 91.1173；用三次样条插值拟合得到的最优点集为{2，9，20，26，33，43，50}，其成本为 93.8561；

用三次多项式拟合得到的最优点集为：{3，11，19，27，35，41，49}，其成本为 132.26665；
用四次多项式拟合得到的最优点集为：{3，9，19，27，35，43，49}，成本为 115.1034；
在使用多项式拟合时，发现另初始化函数 score>120 时，，command window 上没有运行结果，说明程序一直在找满足初始条件的初始点集，但一方面因为多项式拟合并不准确，另一方面数组的变动太小，很难找到成本很小的点集。

5 总结

最佳的拟合方式是选用最佳的特征点和最佳的拟合方程，我们首先通过随机先去点集，通过不同的拟合，比较相应的残差平方均值的大小确定相对较好的拟合方式，有实验结果可得，三次插值拟合更具有代表性，而在多项式拟合中，并不是拟合次数越高，拟合效果越好，在本课题研究中，我们发现在此数据的条件下，四次拟合比其他次项的多项式拟合更具有代表性。得到较好的拟合方程后，我们又通过成本函数的到了能表征样本的最好的 7 个点。其中三次插值得到的点集的成本最小最好，这充分证明了初始研究的的结果，即，三次插值拟合效果最好的结论。
由此可知，在实际的操作中，可以通过三次拟合的方法，采用相应的特征点，检验传感部件的性能，从而达到减少校验产品精度的工作量的目的。故而，统计推断在实际生活中起到了成功的应用。

参考文献

数值分析（第四版）- 李庆扬 -清华大学出版社
上海交大电子工程系 统计推断课程讲义 ftp://202.120.39.248.

附录：

模拟退火算法的程序代码：

```
tic%计算运行时间
clear all
close all
data = xlsread('dataform.xls'); %读入数据;
best_array = zeros (1,7);%定义最优解所在数组
sum_score = 0.0;
best_score = 0.0;
T = 100.0;%初始温度;
score =100;

while score >=35 %找到一个成本小于 35 的点集，减少程序的成本，将该解作为初始解;
   sum_score = 0.0;
   arr1 = randi([1,5]);
   arr2 = randi([6,10]);
   arr3 = randi([11,20]);
   arr4 = randi([21,30]);
   arr5 = randi([31,40]);
   arr6 = randi([41,45]);
   arr7 = randi([46,51]);
   array = [arr1,arr2,arr3,arr4,arr5,arr6,arr7];
   for t = 1:469
       tmpdata=data([2*t-1,2*t],:);
       x_base = tmpdata(1,:);
       y_base = tmpdata(2,:);
       x_temp = x_base(1,array);
       y_temp = y_base(1,array);
       sum_score = sum_score + get_score(x_base,y_base,x_temp,y_temp);
   end
       score = sum_score/469;
       best_score = score;
       best_array = array;
end
%开始模拟退火;
while T > 0.00001
    T = 0.97 * T;
    sum_score = 0.0;
    disp(score);
    %开始调整整个数组的值，先调第一个元素的值;
    while true
        b = array(1);
        c = randi(5);
        if c == 5
            b = b + 1;
        elseif c == 1
            b = b - 1;
        end
        if b >= 1 && b < array(2)%保证产生变动时不会超过范围
            break
        end
    end
    array(1) = b;
    %再调第 2-6 个元素的值;
    for i = 2:6
```

```
        while true
            b = array(i);
            c = randi(5);
            if c == 5
                b = b + 1;
            elseif c == 1
                b = b - 1;
            end
            if b  > array(i - 1) && b < array(i + 1)%保证产生变动时不会超过范围
                break
            end
        end
        array(i) = b;
    end
    %再调第 7 个元素的值；
    while true
        b = array(7);
        c = randi(5);
        if c == 5
            b = b + 1;
        elseif c == 1
            b = b - 1;
        end
        if b > array(6) && b <= 51%保证产生变动时不会超过范围
            break
        end
    end
    array(7) = b;
    disp(array);
    for i = 1:469
       tmpdata = data([2*i-1,2*i],:);
       x_base = tmpdata(1,:);
       y_base = tmpdata(2,:);
       x_temp = x_base(1,array);
       y_temp = y_base(1,array);
       sum_score = sum_score + get_score(x_base,y_base,x_temp,y_temp);
    end

    score = sum_score / 469;
    e = exp(-abs(best_score-score) / T);

    if score <= best_score
        best_score = score;
        best_array = array;

    elseif  rand>e%如果 0 到 1 之间随机数大于概率 e，则数组保持不变；
        array=best_array;
    end

end

disp(best_score);
disp(best_array);

toc
```

评分函数的程序代码;

```
function re = get_score(x_base,y_base,x_temp,y_temp)
score=0;
y_exp = interp1(x_temp,y_temp,x_base,'cubic');
for w = 1:51
    d = abs(y_base(w) - y_exp(w));
        if d <= 0.5
            score = score + 0;
        elseif d <= 1
            score = score + 0.5;
        elseif d <= 2
            score = score + 1.5;
        elseif d<=3
            score = score + 6;
        elseif d <= 5
            score = score + 12;
        elseif d > 5
            score = score + 20;
        end;
end;
score = score + 84;
re=score;
```