

统计推断在数模转换系统中的应用

组号：39 丁天琛 5140309081 王弘奥 5140309245

摘要：本课题的问题来源是为某产品内部的一个测量模块寻求定标工序的优选方案，通过采样进行线性拟合来得到成本较小的解决方案。借助 Matlab，通过比较不同的采样点数量和拟合方式（三次多项式拟合及三次样条插值法），结合遗传算法来取得成本最小的最优解。

关键词：多项式拟合，三次样条插值，遗传算法，Matlab

1 引言

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。

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；传感器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）。

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。

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 ， $i = 1, 2, 3, \dots, 50, 51$ 。

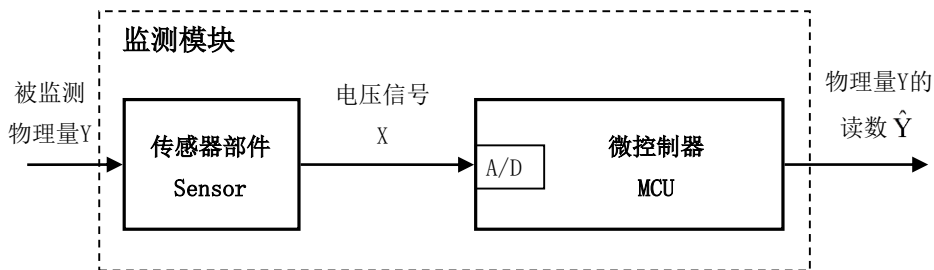


图 1 监测模块组成框图

2 传感模块的定标

2.1 传感部件的特性

一个传感部件个体的输入输出特性大致如图 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。

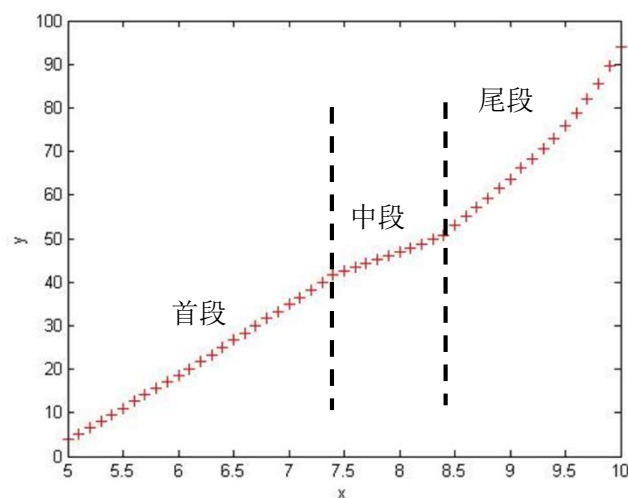


图 2 传感特性图示

2.2 定标方案的评定

为了评估和比较不同方案，现制定以下成本计算规则。

- (1) 单点定标误差成本

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

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 $q=12$ 。某一样本个体的定标成本

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

式中 n_i 表示对该样本个体定标过程中的单点测定次数。

(3) 校准方案总成本

$$C = \frac{1}{M} \sum_{i=1}^M S_i$$

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。

总成本较低的校准方案,认定为较优方案。

3 拟合方案的选取

3.1 采样点数量和分布的确定

拟合曲线是由采样点决定的,因此采样点的数量就至关重要。数量过少会导致拟合曲线精确度下降,从而误差成本增大;数量过多虽然能够提高精确度减小误差成本,但会增加测定成本。因此折衷考虑,我们决定采用种数量的方案来进行比较。

3.2 拟合方法的选取

根据传感部件特性曲线,我们认为三次多项式拟合和三次样条插值拟合可以较好地得到我们需要的结果。因此决定用这两种方法分别进行拟合,比较总成本从而选出最优方案。

3.2.1 三次多项式拟合

我们使用 Matlab 中的 `polyfit` 函数进行多项式拟合。

1. 使用三次多项式对选取的 n 个特征点组合即可能解进行拟合。样本为 $S=\{S_1, S_2, \dots, S_n\}$ 。拟合用方程即为以下方程:

$$u = a_1 d^3 + a_2 d^2 + a_3 d + a_4$$

2. 曲线的拟合度。将 51 个点的输出值代入上述三次多项式计算出相应个输出值,通过评价函数得到评价分值。

3. 实现最优化过程,即不断的寻找存在的可能解,继续前两步骤。每次得到可能解,通过评价,做出一定概率的接受或者舍弃,实现当前解的优化,直至达到终止条件。

3.2.2 三次样条插值

我们使用 Matlab 中的 `spline` 函数进行样条插值。

3.2.2.1 概述

三次样条插值函数法的具体原理如下:

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

- (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 $[a, b]$ 二阶导数连续。

由于 $S(x)$ 在 $[a, b]$ 上二阶导数连续,故在每个节点, $S(x_j-0)=S(x_j+0); S'(x_j-0)=S'(x_j+0); S''(x_j-0)=S''(x_j+0)$ 。共有 $3n-3$ 个条件;除此之外,由插值可得 $n+1$ 个条件,仍需两个,

这两个条件由边值得出：

(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）

周期性便捷条件：即以 $b-a$ 为周期， $S''(x_0+0)=S''(x_n+0), S'(x_0+0)=S'(x_n+0)$ 。这样求出的三次曲线比 2.1 中的多项式拟合应该要精确许多。

4 遗传算法

4.1 遗传算法的介绍

遗传算法（Genetic Algorithm）是模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型，是一种通过模拟自然进化过程搜索最优解的方法。遗传算法的基本运算过程如下：

(1) 初始化：设置进化代数计数器 $t=0$ ，设置最大进化代数 T ，随机生成 M 个个体作为初始群体 $P(0)$ 。

(2) 个体评价：计算群体 $P(t)$ 中各个个体的适应度。

(3) 选择运算：将选择算子作用于群体。选择的目的是把优化的个体直接遗传到下一代或通过配对交叉产生新的个体再遗传到下一代。选择操作是建立在群体中个体的适应度评估基础上的。

(4) 交叉运算：将交叉算子作用于群体。遗传算法中起核心作用的就是交叉算子。

(5) 变异运算：将变异算子作用于群体。即是对群体中的个体串的某些基因座上的基因值作变动。群体 $P(t)$ 经过选择、交叉、变异运算之后得到下一代群体 $P(t+1)$ 。

(6) 终止条件判断：若 $t=T$ ，则以进化过程中所得到的具有最大适应度个体作为最优解输出，终止计算。

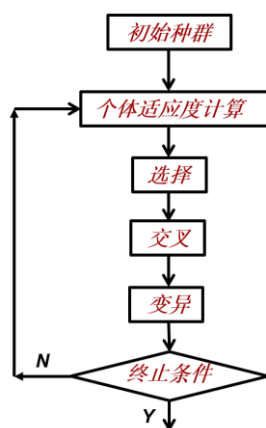


图 3 遗传算法流程图

遗传算法的一般算法：

(1) 建立初始状态：初始种群是从解中随机选取的数据，该种群被称为第一代。

(2) 评估适应度：对每一个解指定一个适应度的值，这个应根据问题求解的实际接近程度来指定。

(3) 繁殖：带有高适应度的个体更可能产生子代。子代是亲代的产物，由来自亲代的基因结合而成，这个过程被称为杂交。同时还有一定的突变概率，会在子代中产生不同于亲代的基因。

(4) 并行计算：非常容易将遗传算法用到并行计算和群集环境中。一种方法是直接把每个节点当成一个并行的种群看待。然后有机体根据不同的繁殖方法从一个节点迁移到另一个节点。另一种方法是“农场主/劳工”体系结构，指定一个节点为“农场主”节点，负

负责选择有机体和分派适应度的值，另外的节点作为“劳工”节点，负责重新组合、变异和适应度函数的评估。

4.2 遗传算法的实现

(1) 初始化：采用随机数算法生成每组 6 或 7 个数共计 500 组来组成一个种群。代数设置为 100 代，变异概率设为 0.1。

(2) 淘汰：在 500 组数据中保留成本较低的前一半（即 250 组），淘汰后一半。

(3) 交叉：在保留下来的 250 组数据中随机两两配对，并交换它们的前后半部分形成新的组合方式加入下一代的循环。

(4) 变异的实现：以 0.1 的概率对每组中的每位数据进行判定，若判定发生变异则将此位数据随机改变成另一个数据。

5 结论

5.1 结果情况

取 6 个点，三次多项式拟合：[2,11,21,31,43,49] 成本：95.5957

取 6 个点，三次样条插值：[3,9,21,31,43,48] 成本：97.3435

取 7 个点，三次多项式拟合：[2,9,26,32,44,51] 成本：96.4593

取 7 个点，三次样条插值：[3,8,19,24,34,44,49] 成本：97.4300

5.2 结果分析

综合上述运行情况，我们发现取点数目为 6 和 7，拟合算法为三次多项式拟合和三次样条插值拟合时最终得到的最小成本解的差别并不是很大。其中当取 6 个点，拟合算法为三次多项式拟合时，能得到成本较低的较优解。

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中的应用课程讲义 [EB/OL] <ftp://202.120.39.248>

[2] 百度百科 三次样条插值[EB/OL] <http://baike.baidu.com/view/2326225.htm>

[3] 百度百科 遗传算法[EB/OL] <http://baike.baidu.com/view/45853.htm>

附录

程序代码：

main.m 文件

```
gen = 100; %代数
change = 0.1; %变异概率
pop_size = 500; %种群规模
pop_num = 6; %每组取点数目
global minput;
minput = dlmread('20150915dataform.csv');
pop = zeros(pop_size, pop_num); %种群初始化
cost_record = zeros(pop_size, 2); %记录成本的数组，第一列为成本，第二列记录排序前的序号
for i=1:pop_size %生成初始的种群
    temp = randperm(51);
    pop(i, :) = temp(1:pop_num);
end
for i=1:gen %遗传主程序
    for j=1:pop_size %计算成本
        cost_record(j, 1) = cost(pop(j, :));
        cost_record(j, 2) = j;
    end
    cost_record = sortrows(cost_record, 1); %按第一列升序排序
    for j=(pop_size/2+1):pop_size
        pop(cost_record(j, 2), 1) = 0; %被淘汰的标记
    end
    cross_queue = randperm(pop_size/2); %生成交叉队列
    t = 1; %空位指针
    for j=1:2:pop_size/2 %两两交叉
        temp1 = zeros(1, pop_num);
        temp2 = zeros(1, pop_num);
        temp1(1:floor(pop_num/2)) =
pop(cost_record(cross_queue(j), 2), 1:floor(pop_num/2));
        temp1(floor(pop_num/2+1):pop_num) =
pop(cost_record(cross_queue(j+1), 2), floor(pop_num/2+1):pop_num);
        temp2(1:floor(pop_num/2)) =
pop(cost_record(cross_queue(j+1), 2), 1:floor(pop_num/2));
        temp2(floor(pop_num/2+1):pop_num) =
pop(cost_record(cross_queue(j), 2), floor(pop_num/2+1):pop_num);
        for k=i:pop_num %变异 temp1
            if(change<rand())
                c = randi(51);
```

```

        while(ismember(c, templ))
            c = randi(51);
        end
        templ(k) = c;
    end
end
for k=i:pop_num %变异 temp2
    if(change<rand())
        c = randi(51);
        while(ismember(c, templ))
            c = randi(51);
        end
        templ(k) = c;
    end
end
while(pop(t,1) ~= 0) %将子代插入空位
    t = t+1;
end
pop(t,:) = templ;
while(pop(t,1) ~= 0)
    t = t+1;
end
pop(t,:) = temp2;
end
end
disp(pop(1,:));
disp(cost_record(1,1));

```

cost.m 文件

```

function result = cost(my_answer)
my_answer_n=size(my_answer,2);
global minput;
[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(nsample,1)*my_answer_n;
result=sum(si)/nsample;

```

mycurvefitting.m 文件

```

function y1 = mycurvefitting( x_premea,y0_premea )
x=[5.0:0.1:10.0];
%y1=interp1(x_premea,y0_premea,x,'spline'); %三次样条插值
y1=polyval(polyfit(x_premea,y0_premea,3),x); %三次多项式拟合
end

```