统计推断在数模转换系统中的应用

第 29 组　　吴昊 5130309339　　刘知威 5130309329

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。以上海交通大学电院电子系实验所测得的共计 469 组输入信号 X 与输出 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 建立函数曲线模型，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，样条插值拟合，遗传算法

The application of statistical reference in the system of AD-DA conversion

ABSTRACT: This report is mainly about the application of statistical inference processing the data in A/D & D/A transformation. By the method of mathematical statistics, with procedures of characteristic point selecting, Algorithm Researching and fitting comparing, relation curve model between the input and the output can be established by Matlab. Finally, we can find a way by which an entire system character will be reflected with less data measuring, thus effectively reduce the engineering design of the cost and improve efficiency.
Key words: sample, feature points, polynomial fitting, Spline difference fitting, genetic algorithm

1 引言

对于一个物理对象的研究可以从其输入输出特性来确定其性质，尤其是目前系统日趋复杂化，从内部结构去研究其性质的复杂性是难以想象。而且在生产实践中，我们真正关心的是输入输出的关系及其产生的影响。有必要提出一种测定输入输出关系的解决方案。在工程实践和科学实验中，为了获得输入与输出的关系,我们往往需要对于一些测定的数据点进行拟合处理，从而推断出变量之间近似的函数表达式关系，但在实际中，我们很难找到非常精确的描述变量之间关系的函数，于是就需要借助统计推断的方法，从已知的数据点中找出能代表变量之间关系的特征点，经过不同的拟合方案得到残差最小的最优解。本次统计推断的研究是为某型产品内部的一个监测模块，寻求校准工序的优化方案。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 具体问题

2.1 研究对象

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。

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）。

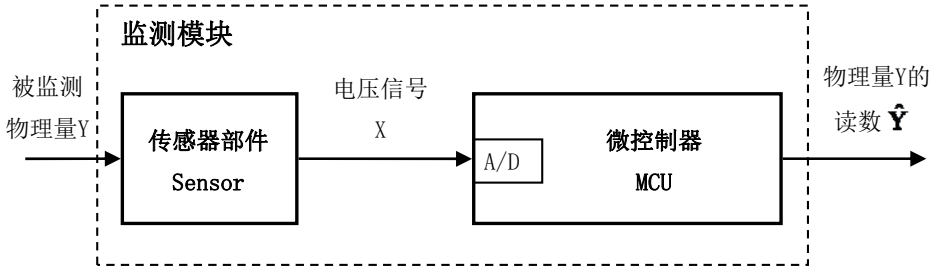

图 2-1 监测模块组成框图

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。

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

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

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

2.2 寻找拟合方程表达式

对于该该检测模块，有相应被检测物理量 Y 与电压信号 X 函数关系，由于该装置的函数关系为非线性，故现需要针对已有的 469 组实验测量数据，对 Y-X 函数关系进行分析，并寻找合适的拟合方法，给出求拟合方程表达式的方法。

3 通过穷举法的拟合实现

3.1 取点数目的分析

就该系统而言，我们注意到：测定 51 组数值后，系统特性已经被充分定标（Calibration），若对每个样本都进行 51 组数值的完整测定，则既费时又高成本，并无直接的工程实用意义。

若能减少需要观测的数值组数量，则可以提高工效。故需要得到一种根据较少的点数就能得到较好的 Y-X 函数关系的一种实用方法。

经过后期多次拟合尝试的结果确定，我们在这里认为取 7 个点为能到达预期效果的保守取点数目。

3.2 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 特性关系做大致的划分，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通过穷举方法来确定特征点的计算量。

3.2.1 基本思路

根据每个系统的Y-X特性关系曲线形状有相似性，且发现其Y-X特性关系曲线大致可分为斜率不同的三个区段，因此可分别在三个区对Y-X的特性曲线进行拟合。拟合时可选用不同方法进行，最后通过计算比较拟合误差来选定拟合方法，进而给出求拟合表达式的方法。

3.2.2 区间划分的确定

区间的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 的范围基本相同，故以 X 为横坐标，绘制 Y-X 图，并通过对 Y 求二次差分并根据二次差分值就能容易分出三个区段的划分点，二次差分离散图如下

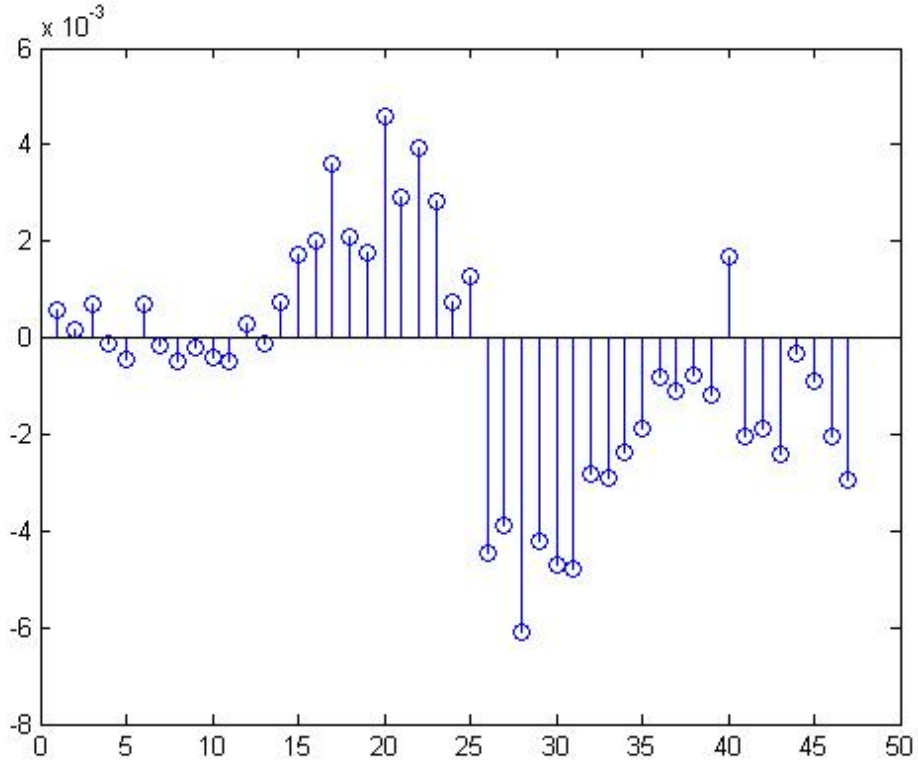

图 3-1 二次差分离散图

3.3 数学模型的建立

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

3.3.1 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

$$P(x)=\sum_{k=0}^{n} a_k x^k \qquad (3\text{-}1)$$

其中$a = x_0 < x_1 < \cdots < x_n = b$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

$$X = \sum_{k=1}^{50} \left[y_k - P(X_k) \right]^2 \qquad (3\text{-}2)$$

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

$$a = ployfit(\chi_0, y_0, m) \qquad (3\text{-}3)$$

进行多项式拟合。其中输入参数x0，y0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

$$\mathbf{y = a_m x^m + L + a_1 x + a_0} \qquad (3\text{-}4)$$

系数$\mathrm{a} = [a_m, L, a_1, a_0]$。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，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。

3.3.2 三次样条插值拟合

三次样条插值拟合，即用6段三次曲线来近似表示。选出7个特征点之后，如右图所示，对于中间的点而言，取中间点加上两侧的与中间点相邻的两个点作为研究对象。通过四个点确定出函数关系方程作为中间点之间曲线方程，以此类推。例如，图中对于S_3、S_6之间而言，由$S_3S_4S_5S_6$确定一条3次曲线，而后仅在S_4S_5之间以该曲线表示，其它点同理。而对于边缘上的点而言，如对点S_1和S_2，由$S_1S_2S_3S_5$确定一条3次曲线，而后仅在S_1S_2之间以该曲线表示。其优点在于它对于相邻两点之间都采用一条三次曲线来表示,这样就能使整体曲线非常平滑,且准确度高。但是该方案的缺点在于每两个点之间都用一条三次曲线拟合,其过程非常复杂， 对于程序的编写和matlab的运算量都是一大考验。

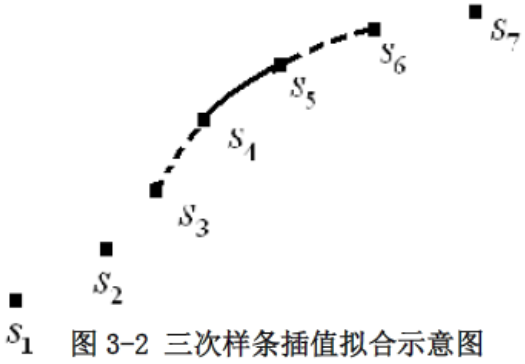

图 3-2 三次样条插值拟合示意图

注： 三次样条插值函数

若函数$s(x) \in C_2[a; b]$($C_2[a; b]$表示区间[a; b]上具有二阶连续导数的函数的全体)，且在每个小区间$[x_j; x_{j+1}]$上是三次多项式，其中$a = x_0 < x_1 < \cdots < x_n = b$是给定节点，则称s(x)是节点$x_0; x_1; \cdots; x_n$上的三次样条函数若在节点$x_j$上给定函数值$y_j = f(x_j)(j=0;1; \cdots ; n)$，并使

$$s(x_j)=y_j(j = 0; 1; \cdots; n) \qquad (3\text{-}5)$$

则称s(x)为f(x)在[a;b]上的三次样条插值函数。

在Matlab 中可使用spline 函数实现。

3.4穷举法特征点的确定

3.4.1 穷举法特征点划分

由于每组数据共有51个采样点，如果直接对这51个点进行拟合，对拟合来说不仅计算量大不适合在实际中对大量的数据进行处理，而且一些误差较大的点会对拟合产生不良影响。故需要找到对拟合来说贡献较大较好的采样点，即特征点。

由二次差分离散图，采样点可被分为3部分，经粗略分析有表3-1：

表3-1区间划分

区间	对应的采样点序号
1	1-15
2	16-25
3	26-51

由图3-1可以看出，区间1线性很好，区间2和区间3变化较大，区间3数值较多，故7个点的选取为，区间1、区间2各2个，区间3取3个。照此方法，一共有$C_{15}^{2}C_{10}^{2}C_{26}^{3}$ = 5386500组。这样可以找到拟合效果最好的7个点，但计算量太大，于是考虑在线性很好的区间1选取点5和9，这样就可以变为$C_{10}^{2}C_{26}^{3}$=51300组，大大减少计算量。

3.4.2特征点的数学评估办法

特征点是拟合效果最好的点，即能使拟合曲线与实际采样点误差最小，为此有如下的数学评估方法：

一、利用统计数学方法对系统特性进行估测

（1）对任意给定的对象装置，先测得一组观测值

$\{(X_i, Y_i)\}$　　其中$i = i_1, i_2, i_3, \ldots, i_P$ ，P<51

（2）利用统计数学方法，估测

$\{(X_j, \widehat{Y_J})\}$　　其中j=1,2,3,...,51

方法一：拟合法

当$i = j$时，允许$\widehat{y_J} \neq y_i$

方法二：插值法

当$i = j$时，必须有$\widehat{y_J} \neq y_i$

二、评价估测的准确程度

（1）增加测量，获得更多的观测值，组成

$\{(X_J, Y_J)\}$　　其中 j=1,2,3,...,51

（2）对比估测值和观测值，给出定量评价

比较$\widehat{Y_J}$和Y_j　　其中 j=1,2,3,...,51

为便于区别和方便说话，我们称上述一中获得的观测值为“事前观测值”，二中增加获得的观测值为“事后观测值”。

三、量化评价估测准确度

为评估和比较不同的校准方案，特制定以下成本计算规则。

- 单点定标误差成本

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

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。

- 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

$$S_i = \sum_{j=1}^{51} s_{i,j} + q \cdot n_i \qquad (3\text{-}7)$$

对样本 i 总的定标成本按式（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。

- 校准方案总体成本

按式（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。

$$C = \frac{1}{M} \sum_{i=1}^{M} S_i \qquad (3\text{-}8)$$

总体成本较低的校准方案，认定为较优方案。

3.5 使用Matlab编程计算

确定特征点的计算过程的算法流程图 3-3：

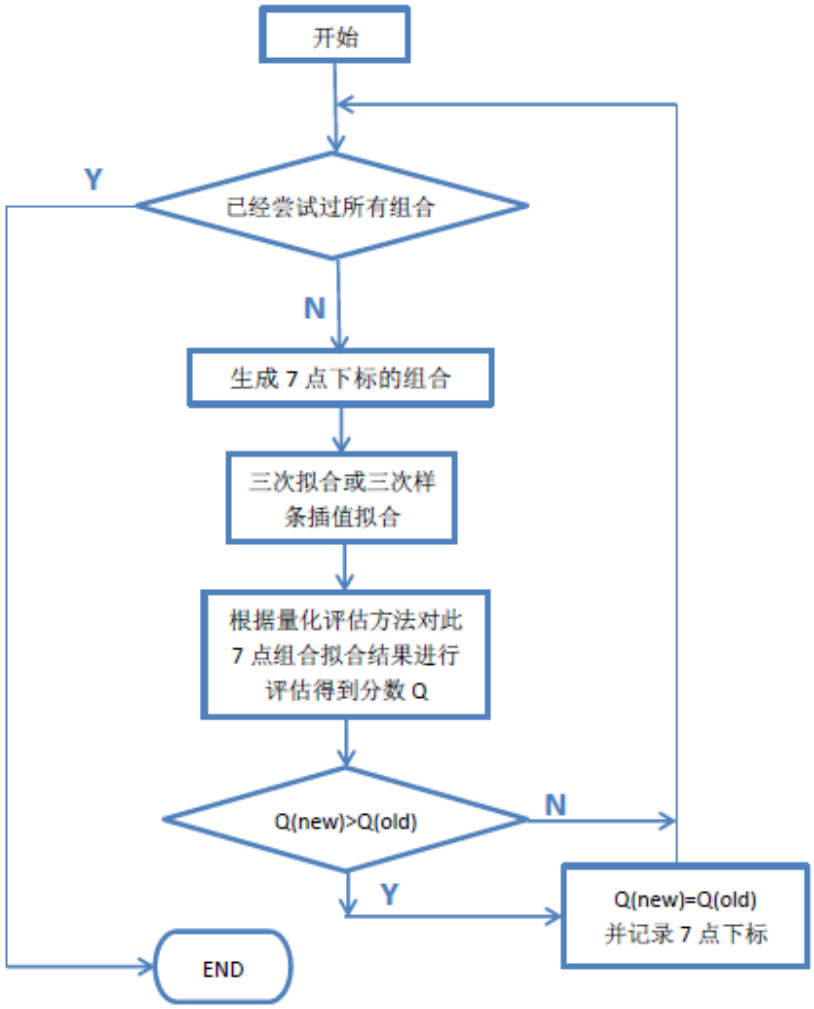

图 3-3 特征点的计算算法流程图

根据流程图利用MATLAB编程计算：

可得到以下结果

表 3-2 最特征点及成本

对穷举点采用的拟合方法	7 个特征点	成本
三次样条插值法	5,9,20,26,33,43,50	97.09
三次多项式拟合	5,9,19,26,34,44,50	141.88

4 遗传算法

4.1 遗传算法简介

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。

它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。所谓交叉是指

把两个父代个体的部分结构加以替换重组而生成新个体的操作。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)。

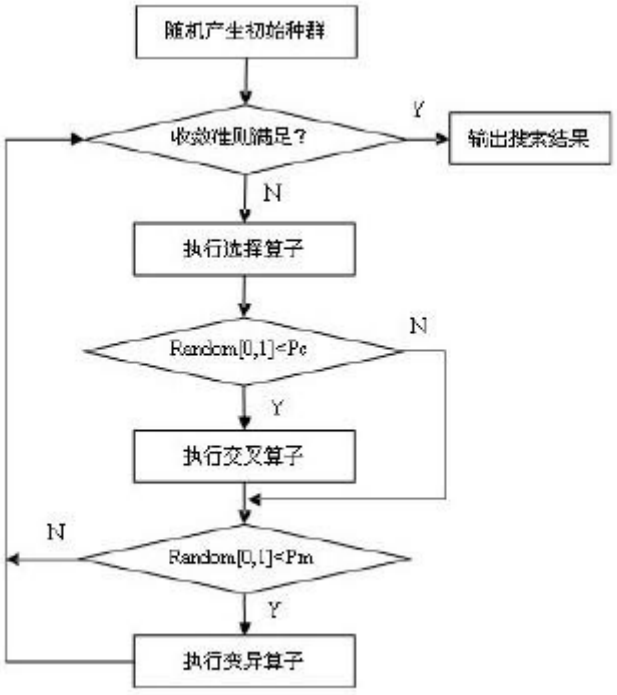

图4-1 遗传算法流程图

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。

4.2 matlab中的运用

1) 编码

遗传算法不对优化问题的实际决策变量进行操作，所以应用遗传算法首要的问题是通过编码将决策变量表示成串结构数据。最常用是采用二进制编码，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。

4.3 实验结果

通过遗传算法用样条插值拟合进行计算，取PopulationSize为50，StallGenLimit为5时：

7个点分别取2　9　20　26　33　43　50时，成本为93.8561；

减少选取的组数，可得：

6个点分别取3　12　22　31　43　50时，成本为92.8774；

5 结论的比较

表 5-1 不同方法所得结果及运行时间的对比

特征点寻找策略	拟合方法		最优特征点	成本	运行时间
适当划分区间后的穷举法	三次样条插值法		5,9,20,26,33,43,50	97.09	3 小时以上
	三次多项式拟合		5,9,19,26,34,44,50	141.88	3 小时以上
遗传算法	三次样条插值法	7 个点	2,9,20,26,33,43,50	93.86	5-10min
		6 个点	3,12,22,31,43,50	92.88	5-10min

通过表5-1的对比可以发现：

1. 用遗传算法产生特征点的效率远远高于适当划分区间后的穷举法产生特征点的效率；
2. 三次多项式拟合的效果十分不理想，可以弃用；
3. 对于采用遗传算法并通过三次样条插值法拟合产生的6个最优特征点的成本是最优的。

通过以上的对比分析可以得到以下结论：

1. 遗传算法在实际应用中可以做到既省时又准确，明显优于穷举法；
2. 适当减少观测点能有效降低成本。

6 结束语

通过对统计推断在模数、数模转换系统中的应用实例的分析，我们的到了一种在工程应用中能够减少测量工作量并快速获得类似于本例中 X-Y 函数的一般方法。即通过遗传算法和样条插值拟合法，求得某个输入输出系统的特征点分布。对于之后的同样系统特性的测量，我们只需要测量器在特征点的输入输出特性，即可获得该系统的输入输出关系。这在工程应用中可以减少不必要的测量，节约时间和生产成本。

7 致谢

感谢袁焱老师的细心指导，让我们了解如何运用统计推断的方法研究实际问题。

8 参考文献

[1]上海交通大学电子工程系统计推断讲座ppt
[2]同济大学数学系计算数学教研室《多项式插值与样条插值》
[3]百度百科遗传算法词条：http://baike.baidu.com/view/45853.htm
[4]百度文库：http://wenku.baidu.com/view/349002020740be1e650e9a94.html

附录

1.二次差分代码：

```
data=csvread('20141010dataform.csv');
a=1:2:(length(data)-1);
b=2:2:length(data);
X=data(a,:);
Y=data(b,:);

for i=2:49
    X1=sum(X(:,i+1))-sum(X(:,i));
    Y1=sum(Y(:,i+1))-sum(Y(:,i));
    T(i-1)=X1/Y1;
end

for i=1:47
dT(i)=T(i+1)-T(i);
end

stem(dT)
```

2.三次多项式拟合：

```
data=csvread('20141010dataform.csv');
a=1:2:(length(data)-1);
b=2:2:length(data);
X=data(a,:);
Y=data(b,:);

Csum=[];
MIN=999999999;  x1=5; x2=9;
for x3=14:24
    for x4=x3+1:25
        for x5=26:48
            for x6=x5+1:49
                for x7=x6+1:50
                x=[x1 x2 x3 x4 x5 x6 x7];
                y1=X(:,x);
                yval=Y(:,x);
               QT=0;
                for q=1:length(a);
                    xcel=y1(q,:);
                    ycel=yval(q,:);
```

```
                    p3=polyfit(xcel,ycel,3);
                    xall=X(q,:);
                    yall=Y(q,:);
                    ycul=polyval(p3,xall);
                     Q=abs(ycul-yall);
                    for j=1:51
                        if Q(j)<=0.5
                            QT=QT+0;
                        else if Q(j)<=1
                                QT=QT+0.5;
                            else if Q(j)<=2
                                    QT=QT+1.5;
                                    else if Q(j)<=3
                                            QT=QT+6;
                                            else if Q(j)<=5
                                                QT=QT+12;
                                                else QT=QT+25;
                                                end
                                        end
                                end
                            end
                        end
                    end
                    QT=QT/length(a);
                    Csum=[Csum QT];
                         if QT<MIN
                            MIN=QT;
                            vfinal=[x1 x2 x3 x4 x5 x6 x7];
                            disp(MIN+84)
                            disp(vfinal);
                         end
                    end

                end
            end
        end
end
```

3. 三次样条插值法

```
data=csvread('20141010dataform.csv');
a=1:2:(length(data)-1);
b=2:2:length(data);
X=data(a,:);
```

```
Y=data(b,:);

Csum=[];
MIN=999999999;    x1=5; x2=9;
for x3=14:24
      for x4=x3+1:25
            for x5=26:48
                  for x6=x5+1:49
                        for x7=x6+1:50
                        x=[x1 x2 x3 x4 x5 x6 x7];
                        y1=X(:,x);
                        yval=Y(:,x);
                       QT=0;
                        for q=1:length(a);
                              xcel=y1(q,:);
                              ycel=yval(q,:);

                              xall=X(q,:);
                              yall=Y(q,:);
                              ycul=spline(xcel,ycel,xall);

                               Q=abs(ycul-yall);
                              for j=1:51
                                    if Q(j)<=0.5
                                          QT=QT+0;
                                    else if Q(j)<=1
                                                QT=QT+0.5;
                                          else if Q(j)<=2
                                                      QT=QT+1.5;
                                                      else if Q(j)<=3
                                                                  QT=QT+6;
                                                                  else if Q(j)<=5
                                                                        QT=QT+12;
                                                                        else QT=QT+25;
                                                                        end
                                                            end
                                                end
                                          end
                                    end
                              end
                        end
                        QT=QT/length(a);
                        Csum=[Csum QT];
                              if QT<MIN
```

```
                                MIN=QT;
                                vfinal=[x1 x2 x3 x4 x5 x6 x7];
                                disp(MIN+84)
                                disp(vfinal);
                            end
                        end

                    end
                end
            end
        end
```

4.六个点的 GA 算法（七个点的代码比较类似故没有给出）

```
%“GA.m”
global data;
data=csvread('20141010dataform.csv');


lb = [ 1, 9, 18, 27, 36, 45];
ub = [ 9, 18, 27, 36, 45, 51];

options = gaoptimset('Generations', 1000,'PopulationSize',50, 'Display', 'iter', 'StallGenLimit',
5);
[x,fval] = ga(@fitness,6,[],[],[],[],lb,ub,[],options);
display(round(x))
display(fval)

%“fitness.m”
function C=fitness(arr1)
    arr2 = [0, 0, 0, 0, 0, 0];
    for x = 1:6
        arr2(x) = int32(arr1(x));
    end
    C = Cost(arr2);
End

%“Cost.m”
function average=Cost(p)
    global data;
    if (p(1)>51||p(2)>51||p(3)>51||p(4)>51||p(5)>51||p(6)>51)
        average = 100000;
        return;
    end
    for i=1:6
```

```
            for j=i+1:6
                if p(i)==p(j)
                    average = 100000;
                    return
                end
            end
        end
        n=469;
        value = 0;
        for k=1:n
            X=data(2*k-1,:);
            Y=data(2*k,:);
            SelectX=[X(p(1)), X(p(2)), X(p(3)), X(p(4)), X(p(5)), X(p(6))];
            SelectY=[Y(p(1)), Y(p(2)), Y(p(3)), Y(p(4)), Y(p(5)), Y(p(6))];
            EstimateY=spline(SelectX,SelectY,X);
            score=evaluate(EstimateY,Y);
            value=value+score;
        end;
        average=value/n;
end

%“evaluate.m”
function c=evaluate(EstimateU,U)
        score=0;
        for n=1:51
            difference=abs(EstimateU(n)-U(n));
            if(difference<=0.5)
                score=score+0;
                continue;
            end
            if(difference<=1)
                score=score+0.5;
                continue;
            end
            if(difference<=2)
                score=score+1.5;
                continue;
            end
            if(difference<=3)
                score=score+6;
                continue;
            end
            if(difference<=5)
                score=score+12;
```

```
            continue;
        end
        if(difference>5)
            score=score+25;
            continue;
        end

    end
    c=score+72;
end
```