

主要参考项目	说明
代码方面	《统计推断在数模模数转换中的应用》，李佳荣，2014 年秋季学期，组号 46 在该组报告附录提供的程序代码基础上，进行了部分修改。
算法描述方面，包含流程图	《统计推断在数模转换系统中的应用》，李佳荣，2014 年秋季学期，组号 46 参考了该组报告的算法描述文字，引用了其流程图。

统计推断在数模转换系统中的应用

----课程报告终稿

组号 01 姓名 张径源 学号 5140309078 ， 姓名 程锦 学号 5140309076

摘要： 本文阐述了本组对课程的认识，以及对定标方案的设想与计划。在设计方案的过程中,我们将主要利用遗传算法 来确定定标方式(取点方式),迭代时通过三次 hermite 插值，进而对单点定标误差成本进行计算，并根据取样点的数目的不同测定成本，最终通过适应度函数法进行筛选，选取遗传算法过程中的最优解。

关键词： 遗传算法，多项式拟合，适应度函数，最优解

Abstract: This paper describes the understanding of the course, as well as the ideas and plans for the program. In the process of design, we mainly use genetic algorithm to determine the calibration method (the method), and then calculate the cost of single point calibration error by three times Hermite interpolation, and then select the optimized sampling scan in the process of genetic algorithm.
Keywords: genetic algorithm, polynomial fitting, fitness function, optimized sampling scan

1. 引言

在产品的生产中我们需要对该产品某模块的输入输出特性进行测量和定标。在定标中，选取的点越多，就越能符合该模块儿输入输出特性的实际情况。但问题在于，在实际生产中，如果选取过多的点定标，所花费的人力、时间等成本也将过高。因此，我们的目的在于如何选取适当的点可以较准确地完成定标工序又不至于定标成本过高。在各种搜索最优解的算法中，我们选择了遗传算法。他通过模拟自然选择的方式完成对最优解的搜索，优点在于他从问题解的串群开始搜索，而不是单个初始解，减少了陷入局部最优解的可能性。其中适应度函数的选择是该算法的核心之一，不同的函数对结果和拟合曲线的相似度都有不同的影响。

2. 符号

表 1 符号约定

X	传感部件的输出电压
Y	传感部件的输入，即被监测物理量

\hat{Y}	监测模块的输出，即监测模块将传感部件输出 x 所转换成的读数（对 Y 的估测值）
$y_{i,j}$	第 i 个样本之第 j 点对应的 Y 实测值（来自标准样本数据库）
$\hat{y}_{i,j}$	第 i 个样本之第 j 点对应的 Y 估测值
$s_{i,j}$	第 i 个样本之第 j 点对应的单点定标误差成本
q	单点测定成本
n_i	对样本 i 定标过程中的单点测定次数
S_i	对样本 i 的定标成本
M	标准样本数据库中的样本总数
C	基于标准样本数据库评价一个校准方案，算得的该方案总体成本

3. 寻找满足条件的拟合函数

3.1 对于样本数据的观察

通过对随机挑选的这 4 组数据的作图和观察，发现曲线明显有两个拐点，作图如下：

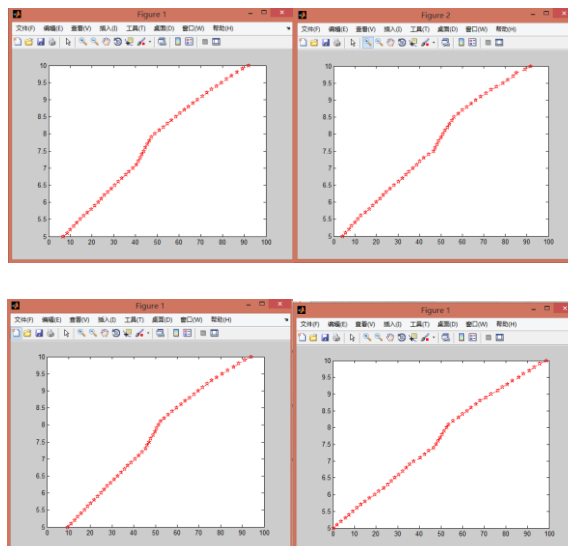


图 1 样品数据绘图实例

3.2 对数据进行 pchip 插值

pchip 指分段三次 hermite 插值。

当 $n=1$ 时，

$$H_3(x)=f(x_0)(1+2(x_0-x)/(x_0-x_1))((x-x_1)/(x_0-x_1))^2+f(x_1)(1+2(x_1-x)/(x_1-x_0))((x-x_0)/(x_1-x_0))^2$$

$$+f'(x_0)(x-x_0)((x-x_1)/(x_0-x_1))^2+f'(x_1)(x-x_1)((x-x_0)/(x_1-x_0))^2$$

在本次课题中，由 matlab 提供的内置函数 `interp`(xx, yx, x, 'pchip') 来完成。

4. 主要数值计算方法

4.1 埃尔米特插值

通常有两种含义。一是指曲线必须通过若干已知离散数据点的一种拟合；二是指利用拟合得到的函数曲线在某区间内的值，作为原本数值未知的点的近似取值。本课程中主要指第二种含义。

对于函数 $f(x)$ ，常常不仅知道它在一些点的函数值，而且还知道它在这些点的导数值。这时的插值函数 $P(x)$ ，自然不仅要求在这些点等于 $f(x)$ 的函数值，而且要求 $P(x)$ 的导数在这些点也等于 $f(x)$ 的导数值。这就是埃尔米特插值问题，也称带导数的插值问题。从几何上看，这种插值要寻求的多项式曲线不仅要通过平面上的已知点组，而且在这些点（或者其中一部分）与原曲线“密切”，即它们有相同的斜率。可见埃尔米特插值多项式比起一般多项式插值有较高的光滑逼近要求。

在本次课题中，三次 hermite 插值由 matlab 提供的内置函数 `interp`(xx, yx, x, 'pchip') 完成。pchip 指分段三次 hermite 插值。这种差值方法可以满足以下要求：1 在节点上的函数值相等。2 对应的导数值相等。3 高阶导数相等。

4.2 遗传算法

遗传算法（Genetic Algorithm）是一类借鉴生物界的进化规律（适者生存，优胜劣汰遗传机制）演化而来的随机化搜索方法。它是由美国的 J.Holland 教授 1975 年首先提出，其主要特点是直接对结构对象进行操作，不存在求导和函数连续性的限定；具有内在的隐并行性和更好的全局寻优能力；采用概率化的寻优方法，能自动获取和指导优化的搜索空间，自适应地调整搜索方向，不需要确定的规则。遗传算法的这些性质，已被人们广泛地应用于组合优化、机器学习、信号处理、自适应控制和人工生命等领域。它是现代有关智能计算中的关键技术。对于一个求函数最大值的优化问题(求函数最小值也类同)。

遗传算法也是计算机科学人工智能领域中用于解决最优化的一种搜索启发式算法，是进化算法的一种。这种启发式通常用来生成有用的解决方案来优化和搜索问题。进化算法最初是借鉴了进化生物学中的一些现象而发展起来的，这些现象包括遗传、突变、自然选择以及杂交等。遗传算法在适应度函数选择不当的情况下有可能收敛于局部最优，而不能达到局最优。

如图所示为其主要流程图。

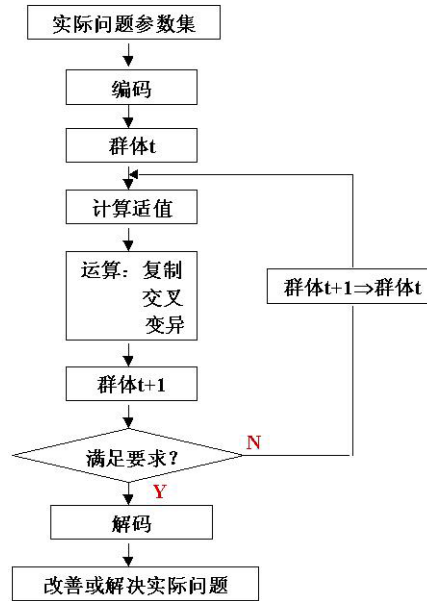


图 2 遗传算法主要流程图

(1) 编码

遗传算法的对象是表示个体的字符串，如果染色体与基因，所以需要把各个数据点表示为统一的字符串，本题采取无符号二进制整数表示。

(2) 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。

(3) 适应度计算

评价每个个体优劣的度，从而决定其遗传机会的大小。其核心是适应度函数，这也是整个遗传算法的关键。适应度函数的选择与具体的问题环境有关，对同一个题，选取不同的适应度函数，最后得到的结果也会不同。在本题中，我们以定标总成本作为其适应度的评价方式。

(4) 选择

选择是从群体中选择优胜个体、淘汰劣质个体的过程。选择常用的方法有适应度比例方法、随机遍历抽样法、局部选择法等。其中最常见的是适应度比例方法，也即轮盘赌选择法。每个个体的生存概率与其适应度成正比，然后通过产生 $[0,1]$ 之间的均匀随机数来选出能产生后代的优质个体。

(5) 交叉

这是产生新个体的主要过程，他以某一概率和方法交换个体间的部分染色体。这里我们采用单点交叉的方法，即对群体随机配对，再随机选择交叉点的位置，最后进行染色体交换。

(6) 变异

变异运算是对个体的某一个或某一些基因座上的基因值按某一较小的概率进行改变，它也是产生新个体的一种操作方法。对于二进制数来说，可以简单地用随机选取变异点，对应值取反来完成。

(7) 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。

5. 成本合理的传感特性校准（定标工序）方案

5.1 设计输入输出函数及采样点个数

选取给定数据中的前几组数据，画出 $Y-X$ 图像，根据初步观察， $Y-X$ 基本符合三次函数关系，故选用三次函数模型对 $Y-X$ 进行定标。确定一个三次函数的参数需要至少 4 个采样点，而根据往届和个人分析经验，9 个采样点已经足以拟合出误差较小的三次多项式，故决定在程序中将采样点个数从 4 到 9 进行依次枚举，得到各个采样点个数下的最小成本方案后，再从其中选出成本最低的一个。

5.2 搜索最优方案

因为数据巨大，所以无法使用穷举法进行搜索。所以我们决定采用启发式搜索的方法来提高搜索效率。

5.2.1 遗传算法的参数确定

我们这里选定种群大小为 50，遗传代数为 50，代沟为 0.7，变异概率为 0.2。为防止种群早熟，我们决定采用较大规模的种群，较多的遗传代数，以及较大的变异概率。

5.3 适应度函数的设计

根据课程设计的要求，采用成本计算规则。

- 单点定标误差成本

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

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

$$S_i = \sum_{j=1}^{51} s_{i,j} + q \cdot n_i \quad (2)$$

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。

- 校准方案总成本

按式（3）计算评估校准方案的总成本，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。

$$C = \frac{1}{M} \sum_{i=1}^M S_i \quad (3)$$

总成本较低的校准方案，认定为较优方案。

5.3.1 单点定标误差成本

在本组初步写的程序中 函数fun1 用来计算某一个个体的定标成本。其中y为实测值，也就是给出的真实数据；y1为通过 函数pchip插值后得到的估测值。根据公式（1）和（2）最后的返回值是 S_i 。

5.3.2 校准方案总体成本

按式（3）计算评估校准方案的总体成本，取所有样本个体的定标成本的统计平均。总体成本较低的校准方案，认定为较优方案。在本组写的程序中成本函数 fitness 可以根据遗传算法得到的取点方案最后得出总成本。

5.3.3 确定校准方案

在完成上述的步骤后，我们确定了大致校准方案。首先枚举采样点个数，在某个确定的采样点个数下，随机生成初始种群，然后根据给定的成本函数值进行挑选，筛选后进行交叉、变异等操作，直至遗传代数达到指定值，此时从当前种群中找到成本最低的一个个体，即为当前采样点个数下的最优校准方案。最后将各校准方案进行比较，最小成本的方案即为最终校准方案。

6. 结论

经过三次多项式拟合，遗传算法迭代，取最大遗传代数为 50，最后可得最佳的取点方案为取 6 个点：4 16 24 31 39 50
取样成本为：86.74

7. 收获

- 【1】初步了解了一种新的程序语言—matlab
- 【2】了解了如何用程序语言来解决实际问题
- 【3】大致了解了统计推断在实际生活中的应用
- 【4】初步了解了遗传算法的应用
- 【5】明白了学习生活中合作的重要性
- 【6】最后也要感谢老师的悉心讲解和解惑

8. 下一步工作

由于时间人力的限制，本次研究成果尚有不足之处。如果有机会，我们可能做以下改进和探索：

- 【1】尝试不同的函数模型对数据点进行拟合或者插值。
- 【2】尝试遗传算法结合退火模拟算法，弥补遗传算法收敛速度慢的缺点。

9. 致谢

感谢袁焱老师的讲座指导，还有助教们的耐心答疑与指导。

10. 参考文献

[1] 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
ftp://202.120.39.248

[2] 百度百科 遗传算法

http://baike.baidu.com/link?url=qrq6T_31hCn44K8VWLWORvIvgSiQ_9jrcnuAzPgg8D6zqInrceXzGX7AHwucMspEq1iSnb2scIMpC4IIQBDCEK

[3] 百度百科 埃尔米特插值

http://baike.baidu.com/link?url=euN06_gcju76mFhfmrErLqRG5MqpiAR3FwwCybfA2z2gRrQeT33TooL331nYQA266eQieWVlIBkEdR6Dtz77

[4] 百度百科 插值

http://baike.baidu.com/link?url=VEo21mYyO6NSW4p-reB2fXV2kApkGBY5mYJBzIkOa6Ble6FsixnQTIGmHS-_pimLKcAvlwRLIU64ZgBDs2Q0csryon-hjt0PzuO2k7HW4re

[5] 《MATLAB 教程》

[6] 《MATLAB 遗传算法工具箱及应用》

附页:

(为简化篇幅, 本清单中未包含遗传工具箱中的函数, 读者可以从网上搜索下载遗传工具箱。)

Main 函数

文件	编辑	导航	断点	运行
----	----	----	----	----

```
1 - A = csvread('20150915dataform.csv');
2
3 % 设定参数
4 - Nind = 50;%个体数目
5 - MAXGEN = 50;%最大遗传代数
6 - GGAP = 0.7;%代沟
7 - min_cost = 1000;
8 - min_sche = zeros;
9 - min_points = 4;
10
11 %min_cost_table 用于记录各采样点下最小成本
12 - min_cost_table = zeros(6,1);
13
14 % 外重循环枚举采样点个数
15 - for num_of_ans = 4:9
16 -     BaseV = crtbase(num_of_ans, 50);
17 -     [Chrom,Lind,BaseV] = crtbp(Nind,BaseV);%初始种群
18 -     l = length(Chrom(:,1));
19 -     for i = 1:l
20 -         Chrom(i,:) = sort(Chrom(i,:));
21 -     end
22 -     ObjV = fitness(A,Chrom);
23 -     tmpbest = Chrom(1,:);
24 -     tmpmincost= ObjV(1,1);
25 -     gen = 0;
26
27 -     ObjV = ObjV';
28 -     while gen < MAXGEN
29 -         FitnV = ranking(ObjV);
```


文件	编辑	导航	断点	运行
----	----	----	----	----

```

30 - SelCh = select('sus', Chrom, FitnV, GGAP);
31 - SelCh = recomb('xovsp', SelCh, 0.7);
32 - SelCh = mut(SelCh, 0.3, BaseV);
33 - ObjVSel = fitness(A, SelCh);
34 - ObjVSel = ObjVSel';
35 - [Chrom, ObjV] = reins(Chrom, SelCh, 1, 1, ObjV, ObjVSel);
36 - [Y, I] = min(ObjV);
37 - % 用上一代最优解替换当前这一代非最优解的个体
38 - if I ~= 1
39 -     Chrom(1,:) = tmpbest;
40 -     ObjV(1,:) = tmpmincost;
41 - end
42 - if I == 1
43 -     Chrom(2,:) = tmpbest;
44 -     ObjV(2,:) = tmpmincost;
45 - end
46 - gen = gen + 1;
47 - fprintf('Num of Points: %d\n', num_of_ans);
48 - fprintf('Generation: %d\n', gen);
49 - fprintf('Min cost : %5.2f\n', min(ObjV));
50 - % 记录这一代的最优解，保留之后用于下一代的替换
51 - [Y, I] = min(ObjV);
52 - tmpbest = Chrom(I,:);
53 - tmpmincost = Y;
54 - end
55 - [Y, I] = min(ObjV);
56 - sche = Chrom(I,:) + 1;
57 - if Y < min_cost
58 -     min_sche = sche;
59 -     min_cost = Y;
60 -     min_points = num_of_ans;
61 - end
62 - min_cost_table(num_of_ans - 3, 1) = Y;
63 - fprintf('The final scheme is(%d points):\n', num_of_ans);
64 - disp(sche);
65 - end
66 - fprintf('The final scheme is:\n');
67 - disp(min_sche);
68 - fprintf('The final cost is :%5.2f\n', min_cost);
69 -

```

fitness 函数:

```
文件      编辑      导航      断点      运行
1  function syd=fitness(A,Chrom)
2  %取点方式为Chrom, n表示对该样本个体定标过程中的单点测定次数。
3      %循环所有的样本。
4
5      L=length(Chrom(:,1));
6      n=length(Chrom(1,:));
7      syd=zeros(1,L);
8
9      for k=1:L
10
11         for i=1:400 %800个样本
12             x = A(i * 2 - 1,:); % x=5:0.1:10;
13             y = A(i * 2 ,:);
14             xdata = zeros(1,n);
15             ydata = zeros(1,n);
16             sort(Chrom(k,:));
17             b = unique(Chrom(k,:));
18             if length(b) < n
19                 syd(1,k) = 469000;
20                 break;
21             end
22             for j = 1:n
23                 xdata(1,j) = x(1,Chrom(k,j)+1 );
24                 ydata(1,j) = y(1,Chrom(k,j)+1 );
25             end
26             %获得得出xdata,ydata 即取点的x和y的一行矩阵。
27             %采用pchip插值
28             y1 = interp1(xdata,ydata,x,'pchip');
29             %调用fun1函数 计算成本。
30
31             syd(1,k)=syd(1,k)+fun1(y1,y,n);
32
33         end
34
35         syd(1,k)=syd(1,k)/400;
36     end
37 end
```

fun1 函数:

	文件	编辑	导航	断点	运行
1	function f=fun1(y1,y,n)%y为实测值，y1为拟合值。成本计算c.				
2	s=0;				
3	for i=1:51%每行有51个数据点				
4	cha=abs(y1(1,i)-y(1,i));				
5	if ((0.4<cha)&&(cha<=0.6))				
6	s=s+0.1;				
7	end				
8	if ((0.6<cha)&&(cha<=0.8))				
9	s=s+0.7;				
10	end				
11	if ((0.8<cha)&&(cha<=1))				
12	s=s+0.9;				
13	end				
14	if ((1<cha)&&(cha<=2))				
15	s=s+1.5;				
16	end				
17	if ((2<cha)&&(cha<=3))				
18	s=s+6;				
19	end				
20	if ((3<cha)&&(cha<=5))				
21	s=s+12;				
22	end				
23	if (5<cha)				
24	s=s+25;				
25	end				
26	end				
27	s=s+12*n;% n表示对该样本个体定标过程中的单点测定次数。				
28	f=s;				
29	end				

draw 函数:

	poly3.m*	fun1.m*	fitness.m*	fitfunction.m	draw.m*	+
1	%绘图函数					
2	function draw()					
3	A = csvread('20150915dataform.csv');%读入数据					
4	%先选定样本1、32、55、122、123					
5	t=[1 32 55 122 123];					
6	for i=t(1,:)					
7	x=A(2*i,:);					
8	y=A(2*i+1,:);					
9	figure, plot(x,y,'rp-.');					
10	end					
11						